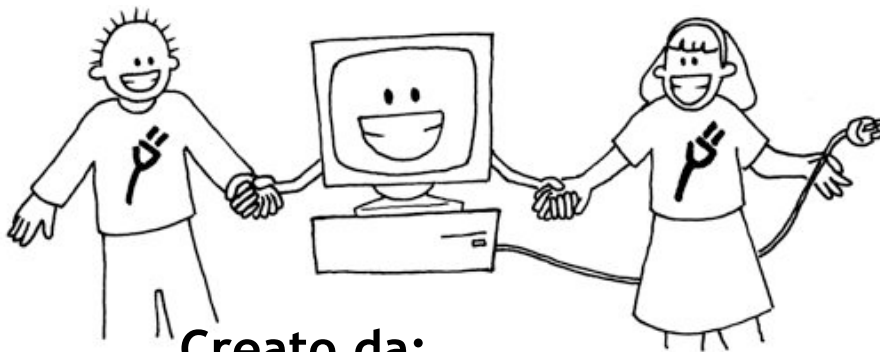


# COMPUTER SCIENCE *Unplugged*

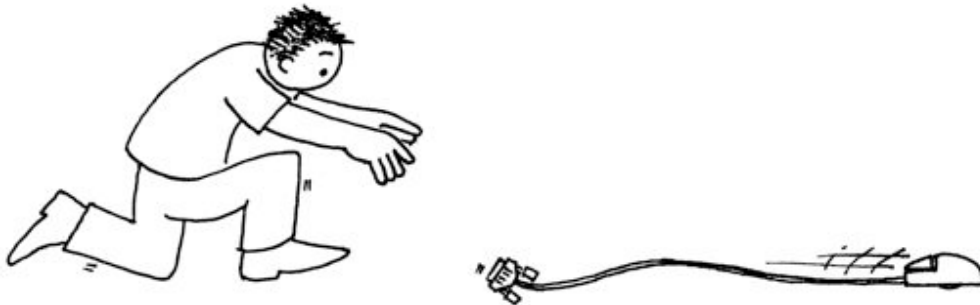
Un testo per imparare l'Informatica  
divertendosi

Per studenti della scuola primaria e  
secondaria inferiore



Creato da:

Tim Bell, Ian H. Witten e Mike Fellows



Adattato per l'uso in classe da  
Robyn Adams e Jane McKenzie

Illustrazioni di Matt Powell

Edizione Italiana a cura di:  
Giovanni Michele Bianco e Renzo Davoli

Ottobre 2009

# Introduzione

---

I Computer sono ovunque. Tutti noi dobbiamo sapere come usarli e molti di noi li usano tutti i giorni. Ma come funzionano? Come *pensano*? E come si fa a renderli più veloci e affidabili, insomma migliori? L'informatica è una scienza affascinante che risponde a queste domande. Le semplici e divertenti attività di questo libro, adatte a bambini e ragazzi di diverse età, introducono i concetti fondamentali dell'informatica, senza che gli studenti debbano usare alcun computer.

Questo libro può efficacemente essere usato in programmi di approfondimento, o anche durante le normali lezioni. Non dovete essere esperti di computer per potervi divertire nell'imparare i principi dell'informatica coi vostri allievi. Il libro comprende diverse attività, con chiare e semplici spiegazioni, risposte alle domande. Ogni attività si conclude con un capitolo denominato "cosa c'entra tutto questo?" che spiega la rilevanza delle attività.

Molte delle attività sono correlate ad argomenti di matematica, per esempio l'esplorazione dei numeri binari, mappe e grafi, problemi di riconoscimento e di ordinamento, crittografia. Altre attività riguardano argomenti di solito trattati in corsi di tecnologia, come per esempio l'apprendimento di come effettivamente funziona un computer. Gli studenti sono coinvolti in attività che sviluppano le capacità di risoluzione di problemi, di comunicazione e la creatività in un contesto significativo ma anche divertente.

Questo libro è stato scritto da tre professori universitari di Informatica e da due insegnanti di scuola ed è basato sulle nostre esperienze in classe. Abbiamo visto che molti importanti concetti di Informatica possono essere insegnati senza un computer. Talvolta un computer è al contrario una distrazione che riduce la capacità di apprendimento. Quindi scollegate (unplug!) i vostri computer e preparatevi ad imparare ciò che veramente è l'Informatica.

**Questo libro è disponibile per il download gratuito per usi personali e didattici grazie a una generosa donazione di Google, Inc. È distribuito secondo le regole della licenza Creative Commons Attribution-NonCommercial-NoDerivs, che significa che siete liberi di copiare, distribuire, mostrare il libro a patto che non facciate modifiche (anche l'indicazione degli autori e questa nota relativa alla licenza deve sempre comparire non modificata); non è possibile usare questo testo per scopi commerciali ed è altresì vietato modificarlo, trasformarlo o usarlo per altre attività o pubblicazioni. Noi incoraggiamo l'uso di questo testo per scopi didattici, e per questo fine siete autorizzati a stampare la vostra copia del libro e a distribuire i fogli di lavoro ai vostri studenti. Tutte le richieste e i suggerimenti sono benvenuti, potete inviarli direttamente agli autori come indicato nel sito [www.csunplugged.org](http://www.csunplugged.org).**

**Questo libro è stato tradotto in molte lingue. Potete verificare nel sito web la disponibilità delle traduzioni del testo.**

# Ringraziamenti.

---

Molti ragazzi e molti insegnanti ci hanno aiutato a perfezionare le nostre idee. Gli studenti e gli insegnanti della South Park School (Victoria, BC), Shirley Primary School, Ilam Primary School and Westburn Primary School (Christchurch, New Zealand) sono state le nostre cavie per molte attività. Siamo particolarmente grati a Linda Picciotto, Karen Able, Bryon Porteous, Paul Cathro, Tracy Harrold, Simone Tanoa, Lorraine Woodfield, and Lynn Atkinson per averci accolto nelle loro classi e per i loro preziosi suggerimenti per il perfezionamento delle attività. Gwenda Bensemman ha sperimentato molte delle attività per noi e ci ha suggerito modifiche. Richard Lynders and Sumant Muruges h ci hanno aiutato nei tentativi in classe. Parte delle attività di crittografia sono state sviluppate da Ken Noblitz. Alcune attività sono state portate avanti sotto l'ombrello del gruppo denominato Victoria "Mathmania", con l'aiuto di Kathy Beveridge. Le prime versioni delle illustrazioni sono state fatte da Malcolm Robinson and Gail Williams, e ci siamo anche avvalsi dei consigli di Hans Knutson. Matt Powell ha anche fornito una preziosa assistenza durante lo sviluppo del progetto "Unplugged". Siamo grati alla Brian Mason Scientific e alla Technical Trust per i generosi contributi durante le prime fasi di sviluppo di questo libro.

Un ringraziamento speciale va a Paul e Ruth Ellen Howard, che hanno sperimentato molte delle attività e fornito numerosi utili suggerimenti. Anche Peter Henderson, Bruce McKenzie, Joan Mitchell, Nancy Walker-Mitchell, Gwen Stark, Tony Smith, Tim A. H. Bell<sup>1</sup>, Mike Hallett, e Harold Thimbleby hanno fornito molti utili suggerimenti.

Abbiamo un enorme debito con le nostre famiglie: Bruce, Fran, Grant, Judith, and Pam for their support, e Andrew, Anna, Hannah, Max, Michael, e Nikki che ci hanno ispirato una grande parte di questo lavoro,<sup>2</sup> e che spesso sono stati i primi a sperimentare le attività.

Siamo particolarmente riconoscenti a Google Inc. per aver finanziato il progetto Unplugged, e per averci consentito di realizzare questo testo disponibile per l'uso gratuito.

Tutte le richieste e i suggerimenti sono benvenuti, potete inviarli direttamente agli autori come indicato nel sito [www.csunplugged.org](http://www.csunplugged.org).

---

<sup>1</sup> Non ci sono relazioni di parentela col primo autore.

<sup>2</sup> Infatti l'attività relativa alla compressione del testo è stata inventata da Michael.

# Indice

---

Introduzione.....	i
Ringraziamenti.....	ii
Indice.....	iii
<b>Parte I.....</b>	<b>1</b>
<b><u>I dati: la materia prima per rappresentare le informazioni.....</u></b>	<b><u>1</u></b>
I dati: la materia prima.....	2
Attività 1.....	3
Conta i punti – I numeri binari.....	3
I numeri binari.....	4
Pagina da fotocopiare: numeri binari .....	8
Cosa c'entra tutto questo?.....	16
Soluzioni e suggerimenti.....	18
Attività 2.....	19
Colorare coi numeri —La rappresentazione delle immagini.....	19
Colorare coi numeri.....	20
Cosa c'entra tutto questo?.....	27
Soluzioni e suggerimenti.....	28
Attività 3.....	29
Puoi dirlo nuovamente! —La compressione del testo.....	29
Puoi dirlo nuovamente!.....	30
Cosa c'entra tutto questo?.....	36
Soluzioni e suggerimenti.....	36
Attività 4.....	37
La magia delle carte girate—Il riconoscimento e la correzione degli errori.....	37
Il “trucco magico”.....	38

Cosa c'entra tutto questo?.....	43
Soluzioni e suggerimenti.....	44
Attività 5.....	45
Indovina indovinando — La Teoria dell'Informazione.....	45
Indovina indovinando.....	46
Attività dell'indovina indovinando.....	48
Cosa c'entra tutto questo?.....	51
Soluzioni e suggerimenti.....	52
<b>Parte II.....</b>	<b>53</b>
<b><u>Far lavorare i computer—Algoritmi.....</u></b>	<b><u>53</u></b>
Far lavorare i computer.....	54
Attività 6.....	55
Battaglia navale—Algoritmi di ricerca.....	55
Battaglia navale.....	56
Battaglia navale — Un gioco a ricerca lineare.....	57
Battaglia navale — Un gioco a ricerca binaria.....	58
Battaglia navale — Un gioco a ricerca Hash.....	59
Attività aggiuntive.....	60
Cosa c'entra tutto questo?.....	73
Attività 7.....	74
Il piccolo e il grande—Algoritmi di ordinamento.....	74
Il piccolo e il grande.....	75
Cosa c'entra tutto questo?.....	81
Soluzioni e suggerimenti.....	82
Attività 8.....	83
Batti il tempo—Reti di ordinamento.....	83
Reti di ordinamento.....	84

Foglio da fotocopiare: reti di ordinamento.....	85
Cosa c'entra tutto questo?.....	88
Attività 9.....	89
La città fangosa—Minimal Spanning Trees.....	89
La città fangosa.....	90
Cosa c'entra tutto questo?.....	94
Soluzioni e suggerimenti.....	94
Attività 10.....	95
Il gioco dell'arancia—Instradamento e deadlock nelle reti.....	95
Il gioco dell'arancia.....	96
Cosa c'entra tutto questo?.....	98
<b>Parte III.....</b>	<b>99</b>
<b><u>Dire ai computer cosa devono fare—Rappresentare le procedure.....</u></b>	<b><u>99</u></b>
Dire ai computer cosa devono fare.....	100
Attività 11.....	101
Caccia al tesoro—Automati a stati finiti.....	101
L'Isola del Tesoro.....	102
Carte per l'esempio pratico.....	104
Carte per l'esempio pratico.....	105
Cosa c'entra tutto questo?.....	115
Soluzioni e suggerimenti.....	117
Attività 12.....	118
Gli ordini di marcia—I linguaggi di programmazione.....	118
Ordini di marcia.....	120
Cosa c'entra tutto questo?.....	122





# **Parte I**

**I dati: la materia prima per rappresentare le informazioni.**

# I dati: la materia prima.

---

## Come vengono immagazzinate le informazioni nei Computer?

La parola “computer” viene dal latino *computare*, che significa calcolare o sommare insieme, ma i computer di oggi sono molto di più di giganti calcolatrici! Possono essere biblioteche, ci aiutano a scrivere, a cercare informazioni, ci fanno ascoltare brani musicali o vedere film. Ma come fanno a immagazzinare, a *memorizzare*, tutte queste informazioni? Che ci crediate o no, il computer usa solo due oggetti: lo zero e l'uno.

## Qual è la differenza fra dati e informazioni?

I dati sono materie prime grezze, i numeri con i quali il computer lavora. Un computer trasforma i dati in informazioni (parole, numeri, immagini, suoni) che potete comprendere.

## Come si possono convertire numeri, lettere, parole, figure, suoni in zero e uno?

In questa parte del libro impareremo cosa sono i numeri binari, come i computer disegnano le figure, come funziona un fax, quale sia il modo efficiente per memorizzare tanti dati, come evitare gli errori di memorizzazione, e come possiamo misurare la quantità di informazione che tentiamo di immagazzinare.



# Attività 1

---

## Conta i punti - *I numeri binari*

### Sommario

I computer rappresentano e trasferiscono i dati come sequenze di zero e uno. Come possiamo rappresentare parole e numeri usando solamente questi due simboli?

### Competenze richieste:

Gli studenti devono essere in grado di:

- ✓ Contare
- ✓ Creare corrispondenze
- ✓ Mettere in sequenza

### Età

- ✓ A partire da 7 anni

### Materiale

- ✓ Serve un set di cinque carte (vedi pagina 7) per la spiegazione. In alternativa possono essere utilizzati fogli A4 con adesivi di smiley (faccine sorridenti) come punti.

Ogni studente deve avere:

- ✓ Un set di cinque carte.  
Fotocopiate pagina 7 su di un cartoncino e ritagliate le carte.
- ✓ Il foglio di lavoro “numeri binari” (pagina 6)

Sono previste anche attività di approfondimento opzionali, per le quali ogni studente deve avere:

- ✓ Il foglio di lavoro: “lavorare coi numeri binari” (pagina 8)
- ✓ Il foglio di lavoro: “spedire messaggi segreti” (pagina 9)
- ✓ Il foglio di lavoro: “i fax e i modem” (pagina 11)
- ✓ Il foglio di lavoro: “contare oltre il numero 31” (pagina 12)
- ✓ Il foglio di lavoro: “ancora di più sui numeri binari” (pagina 14)

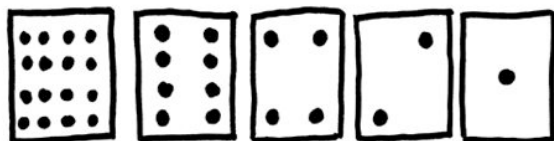
# I numeri binari

---

## Introduzione

Prima di distribuire il foglio di lavoro di pagina 6 è utile che mostriate le regole del gioco a tutta la classe.

Per questa attività, avete necessità di cinque carte, come mostrato qui sotto, con punti su un lato e niente sull'altro. Scegliete cinque studenti che si pongano uno a fianco all'altro di fronte al resto della classe. Date a ognuno di loro una delle carte in modo che appaiano alla classe nell'ordine seguente:



## Discussione

Quale regola unisce il numero dei punti presente sulle carte? (Ogni carta ha il doppio dei punti della carta immediatamente alla destra).

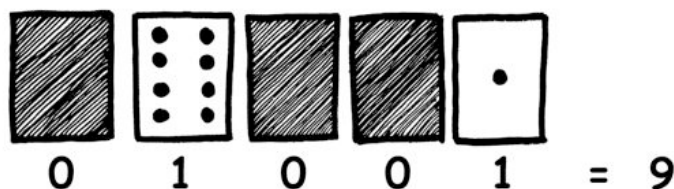
Quanti punti avrebbe la prossima carta se ne aggiungessimo una a sinistra? (32) e quella successiva?...

Potete usare queste carte per scrivere numeri tenendone alcune coi punti rivolti verso la classe e girando le rimanenti dal lato del dorso. La somma dei punti visibili è il numero. Chiedete agli studenti di scrivere 6 (la carta 4 e la carta 2) poi 15 (la 8, la 4, la 2 e la 1), quindi 21 (16, 4 e 1)...

Ora provate a contare partendo da 0

Il resto della classe deve osservare attentamente come le carte cambino per vedere se riescono a trovare una regola su quando ogni carta venga girata (ogni carta viene girata la metà delle volte di quella alla sua destra). Magari provate questo esercizio con più di un gruppo di cinque studenti.

Una carta visibile, cioè esposta dal lato coi punti, si rappresenta con un uno. Una carta girata dal lato senza punti, si rappresenta con uno zero. Questo è il sistema di numerazione binario.



Chiedete agli studenti di calcolare quale numero sia 01001 in decimale (9) e quale numero binario corrisponda al decimale 17 (10001)

Fate ancora altri esercizi per essere certi che tutti abbiano capito.

Ci sono cinque attività opzionali per l'approfondimento. Gli studenti dovrebbero svolgerne più che possono.

# Foglio di lavoro: i numeri binari

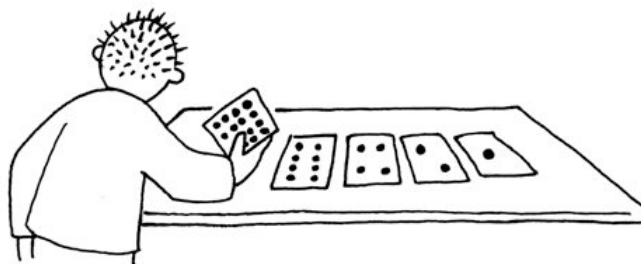
## Imparare a contare

Pensavate di saper contare? Bene, ora imparerete un nuovo modo per farlo!

Sapevate che i computer usano solo zero e uno? Tutto quello che vedete o sentite dal vostro computer, parole, immagini, numeri, filmati e anche suoni sono immagazzinati usando solo questi due numeri! Queste attività vi spiegheranno come mandare messaggi segreti ai vostri amici usando lo stesso metodo dei computer.

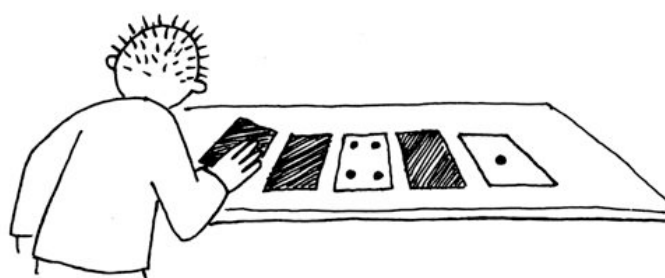
## Istruzioni

Ritagliate le carte dal vostro foglio o ponetele davanti a voi in modo che la carta con 16 punti sia alla vostra sinistra come vedete qui:



Assicuratevi che le carte siano esattamente poste nell'ordine indicato nella figura.

Ora, mantenendo le carte nello stesso ordine rovesciate alcune carte sul dorso in modo che siano visibili esattamente 5 punti.

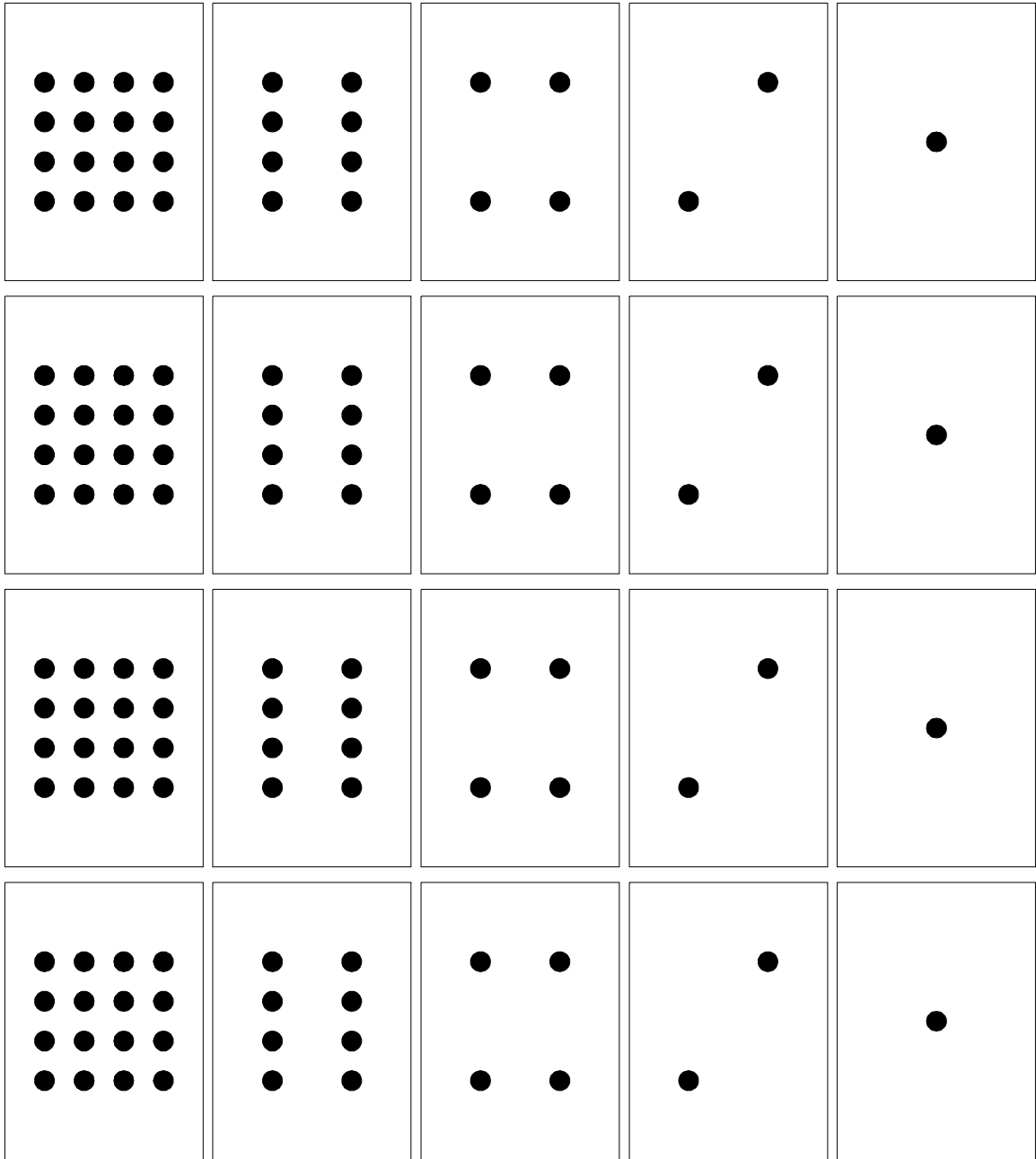


Nello stesso modo provate ora a far comparire 3 punti, poi 12 e 19. Quanti modi possibili esistono per far comparire ogni numero? Qual è il massimo numero di punti che riuscite a far comparire? Qual è il minimo? C'è qualche numero di punti che non potete ottenere fra il numero minimo e quello massimo?

**Extra per esperti:** Provate con i numeri 1, 2, 3, 4 in ordine. Potete descrivere un metodo per aggiungere un punto alla volta?

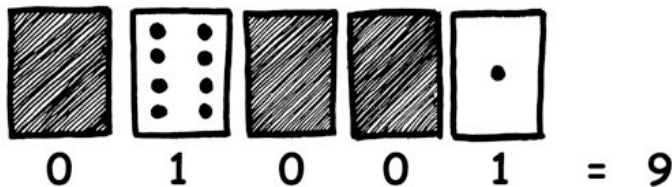
# Pagina da fotocopiare: numeri binari

---



## Foglio di lavoro: lavorare coi numeri binari

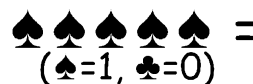
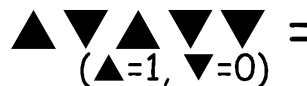
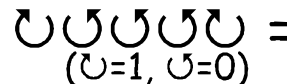
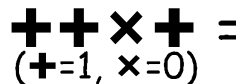
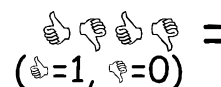
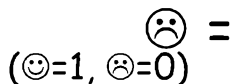
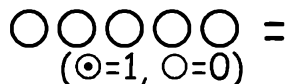
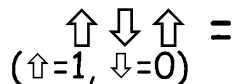
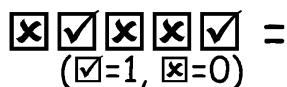
Il sistema binario usa i numeri **zero** e **uno** per rappresentare quali carte sono dal lato con i punti e quali sul dorso. **0** corrisponde a una carta sul dorso e **1** significa che potete vedere i punti. Per esempio:



Potete scoprire cosa è **10101**? E **11111** a quale numero corrisponde?

In quale giorno e in quale mese siete nati? Scrivetelo in binario. Scoprite come il giorno del compleanno dei vostri amici si scrive in binario.

### Provate ora a scoprire questi numeri in codice:

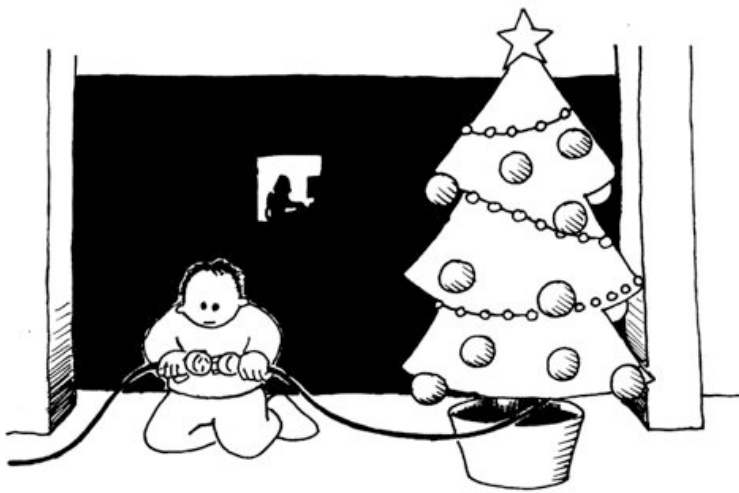


**Extra per esperti:** Usando una serie di regoli di lunghezza 1, 2, 4, 8 e 16 unità mostrate come potete giustapporli per realizzare sequenze di ogni lunghezza fino a 31 unità. Oppure con una bilancia a due piatti e cinque pesi da 1, 2, 4, 8, 16 unità, si possono misurare pesi da 0 unità a 31 unità.

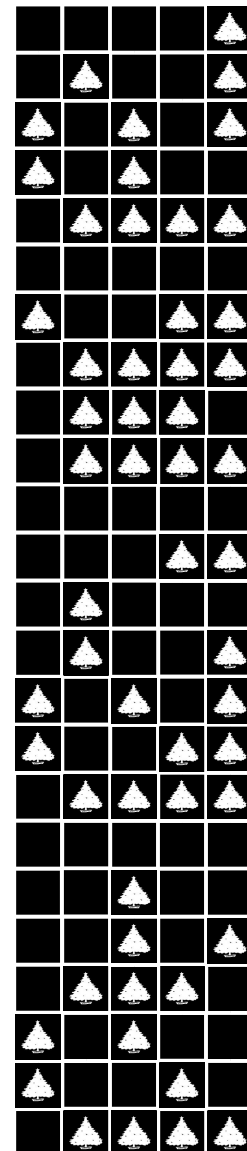


## Foglio di lavoro: inviare messaggi segreti

Tom è intrappolato all'ultimo piano di un grande magazzino. Mancano pochi giorni a Natale e voleva tornare a casa con alcuni regali ma non si è accorto dell'orario di chiusura ed è rimasto all'interno. Ha provato a chiamare, a urlare ma non c'è nessuno. È ormai notte, Tom vede nel palazzo di fronte una ragazza specialista in informatica che sta lavorando. Come può attirare la sua attenzione? Tom si guarda attorno per vedere cosa possa usare. Ha una idea geniale! Può usare le luci dell'albero di Natale per mandarle un messaggio! Tom connette le luci in modo da poterle accendere e spegnere tutte insieme. Quindi usa un semplice codice binario che è sicuro che la ragazza dall'altro lato della strada è capace di comprendere. Volete provarci anche voi?



1	2	3	4	5	6	7	8	9	10	11	12	13
a	b	c	d	e	f	g	h	i	j	k	l	m
14	15	16	17	18	19	20	21	22	23	24	25	26
n	o	p	q	r	s	t	u	v	w	x	y	z



## Foglio di lavoro: posta elettronica e modem

Anche i computer collegati a Internet attraverso un modem usano il sistema binario per spedire i messaggi. La sola differenza è che loro usano dei suoni. Un "bip" acuto viene usato per spedire un uno mentre uno di tono basso per uno zero. Questi suoni sono molto veloci, così veloci che i vecchi modem attaccati alla presa del telefono durante la fase di collegamento facevano udire solamente un fastidioso gracchiare. Se non l'avete mai sentito, provate a chiamare un fax, le macchine del fax usano questo tipo di modem per spedire informazioni.



Utilizzando lo stesso codice usato da Tom per chiedere aiuto nel grande magazzino, provate a mandare un messaggio di posta elettronica a un vostro amico o a una vostra amica. Non dovete essere veloci come i veri modem, altrimenti non riuscirete a capire il messaggio!



## Foglio di lavoro: contare oltre il numero 31

Osservate ancora la sequenza delle carte dei numeri binari. Se doveste creare la prossima carta della sequenza quanti punti dovrebbe avere? E quale è la carta ancora successiva? Quale è la regola che seguite per creare le nuove carte? Come potete osservare poche carte sono necessarie per contare fino a numeri molto grandi.

Se osservate la sequenza, potrete osservare una interessante relazione:

**1, 2, 4, 8, 16...**

Provate a sommare:  $1 + 2 + 4 = ?$  Qual è il risultato?

E ora provate:  $1 + 2 + 4 + 8 = ?$

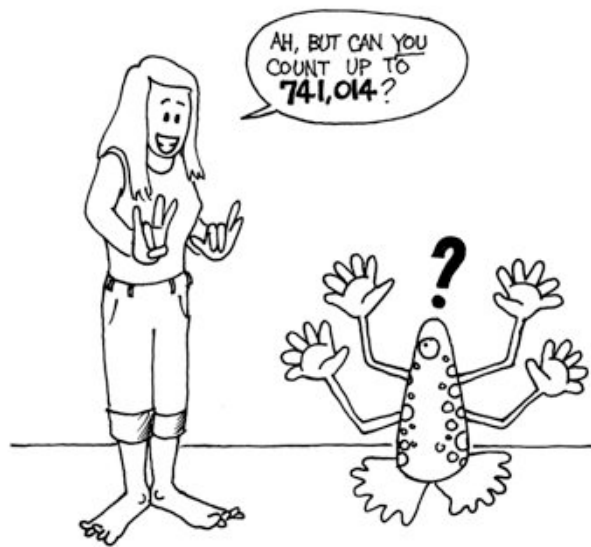
Cosa succede se sommate tutte le carte?

Ora proviamo a contare con le dita, non nel modo solito ma usando il binario. Voi siete normalmente capaci di contare fino a dieci? Bene con il binario potete contare fino a numeri molto più grandi di dieci, senza la necessità di essere esseri alieni con tante dita! Usando i numeri binari potete contare da 0 a 31 usando una mano sola facendo in modo che ogni dito rappresenti una delle carte. Il pollice corrisponde alla carta con un punto, l'indice a quella con 2, il medio a quella con 4 e così via. Potete rappresentare con una mano 32 diversi numeri (non dimenticate che anche lo zero è un numero!)

Provate allora a contare usando le dita, il dito alzato rappresenta uno, cioè è come se la carta corrispondente fosse visibile, se il dito è abbassato è uno zero.

Usando entrambe le mani potete contare da 0 a 1023! Sono 1024 numeri!

Se avete le dita dei piedi snodate (e qui dovrete veramente essere alieni) potreste ottenere numeri molto più alti. Se con una mano potete ottenere 32 numeri, con due mani  $32 \times 32 = 1024$  numeri, fino a quale numero può contare la signorina "dita-dei-piedi-snodate" che conta in binario con tutte e 20 le dita delle mani e dei piedi?



(e tu sai contare fino 741.014?)

## Foglio di lavoro: ancora di più sui numeri binari.

1. Un'altra interessante proprietà dei numeri binari la potete osservare quando aggiungete a un numero uno zero dopo l'ultima cifra a destra. Se lavorate con i numeri in base 10 (i numeri decimali), quando aggiungete uno zero moltiplicate il numero per 10. Per esempio 9 diventa 90, 30 diventa 300.

Ma cosa succede quando aggiungete uno zero alla dell'ultima cifra di un numero binario? Provate con questo esempio:

$$\begin{array}{ccc} 1001 & \parallel & 10010 \\ (9) & & (?) \end{array}$$

Fate le vostre ipotesi. Quale è la regola? Perché secondo voi accade questo?

2. Ogni carta che abbiamo usato rappresenta un "bit" del computer (infatti bit è l'abbreviazione di "**b**inary **d**igit" che in inglese significa semplicemente cifra binaria). Il nostro codice per rappresentare l'alfabeto usa solo cinque carte cioè cinque bit. I computer usano più simboli, devono riconoscere le lettere minuscole da quelle maiuscole, quelle accentate, i numeri, i simboli della punteggiatura, i simboli speciali come \$ oppure %.

Osservate una tastiera e provate a calcolare quanti caratteri compongono l'alfabeto che un computer deve rappresentare. Quanti bit sono necessari per un computer per poter rappresentare tutti questi simboli?

La maggior parte dei computer oggi usa una rappresentazione standard chiamata ASCII (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange, tradotto suona come "codice americano standard per lo scambio di informazioni"), che è basato su questo numero di bit per ogni carattere. Questo codice, nato per la lingua inglese, va bene anche per l'italiano, il francese, il tedesco, lo spagnolo, ... ma in alcune lingue hanno molti più simboli e quindi hanno necessità di codici composti da più bit.



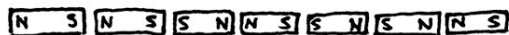
# Cosa c'entra tutto questo?

---

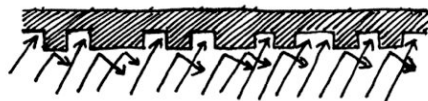
I computer oggi usano il sistema binario per rappresentare l'informazione. È chiamato binario perché usa due sole cifre. Viene anche chiamato base due (gli umani di solito usano il sistema decimale cioè la base dieci). Ogni cifra (zero o uno) viene chiamata bit (**binary digit**, cioè cifra binaria). Un bit è rappresentato nella memoria principale di lavoro di un computer come un transistor che è acceso o spento o da un condensatore carico o non carico.



Quando i dati devono essere trasmessi su di una linea telefonica, o attraverso un canale radio, vengono usati toni di diversa frequenza, acuti o bassi per rappresentare cifre uno o zero. Sui dischi magnetici (floppy o dischi rigidi) e sui nastri i bit sono rappresentati come direzioni del campo magnetico su una superficie ricoperta di materiale magnetizzabile. I due valori sono memorizzati con elementi magnetizzati in direzione nord-sud o sud-nord.



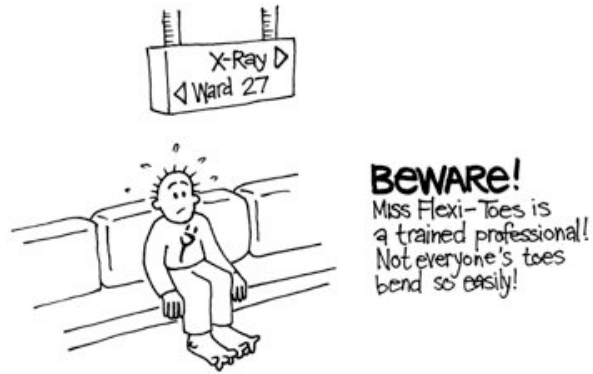
I CD musicali, i CD-ROM e i DVD registrano i bit in modo ottico: le parti della superficie corrispondenti a un bit riflettono o no la luce.



Un bit non può rappresentare molto, ma i bit sono normalmente considerati otto a otto. Ogni gruppo di otto bit, denominato byte, può rappresentare numeri da 0 a 255.

La velocità di un computer dipende anche da quanti bit può elaborare contemporaneamente. Per esempio molti dei computer oggi si dicono "a 32 bit" perché elaborano a ogni passo numeri fino 32 bit. I computer a 16 bit, quando devono elaborare numeri a 32 bit devono spezzare i dati e fare più operazioni e quindi sono più lenti.

I bit e i byte sono tutto ciò che il computer usa per memorizzare e trasmettere numeri, testo, e tutte le altre informazioni. In alcune attività, più avanti nel testo, vi mostreremo come diversi tipi di informazione possano essere rappresentati da computer.



(sala di attesa per i raggi X: “Attenzione, la signorina dita-dei-piedi-snodate è una contorsionista professionista. Non tutti possono piegare le dita dei piedi così facilmente!”)



# Soluzioni e suggerimenti

---

Numeri binari (pagina 6)

**3** si rappresenta con le carte 2 e 1

**12** si rappresenta con le carte 8 e 4

**19** si rappresenta con le carte 16, 2 e 1

C'è una sola rappresentazione per ogni numero.

Il massimo numero rappresentabile è 31, il più piccolo è 0. È possibile rappresentare tutti i numeri naturali compresi fra questi estremi e tutti hanno una rappresentazione unica.

**Extra per esperti:** Per incrementare il numero di uno occorre girare ogni carta a partire da destra fino a quando una carta viene girata dal dorso al recto (se la carta che mostrava i punti viene coperta si continua, se invece una carta che era coperta mostra ora i punti il procedimento termina).

Lavorare coi numeri binari (pagina 9)

10101 = 21, 11111 = 31

Inviare messaggi segreti (pagina 10)

Ecco il messaggio spedito da Tom: AIUTO SONO CHIUSO DENTRO

Contare oltre il numero 31 (pagina 12)

Se sommate tutti i numeri sulle carte la somma sarà sempre di una unità minore del prossimo numero nella sequenza.

La signorina dita-dei-piedi-snodate può rappresentare  $1024 \times 1024 = 1,048,576$  numeri— può contare da 0 a 1,048,575!

Ancora di più sui numeri binari (pagina 14)

Quando viene aggiunto uno zero il numero rappresentato viene raddoppiato.

Tutti gli uno del numero (i punti visibili delle carte) assumono una posizione che rappresenta un valore doppio (occorre cambiare ogni carta con quella alla sinistra), il valore totale viene quindi duplicato. (In base 10 aggiungere uno zero alla destra equivale a moltiplicare per 10, la base).

Un computer ha necessità di 7 bit per registrare tutti i caratteri. Questo consente fino a 128 caratteri diversi. Di solito si usano 8 bit con un bit inutilizzato.

# Attività 2

---

## Colorare coi numeri –*La rappresentazione delle immagini*

### Sommario

I computer memorizzano i disegni, le fotografie e le altre figure usando solo numeri. Questa attività mostra come fanno.

### Competenze richieste

- ✓ Contare
- ✓ Disegnare

### Età

- ✓ A partire da 7 anni

### Materiale

- ✓ trasparenze da proiettore ottenute fotocopiando il prototipo “colorare coi numeri” (pagina 21)

Ogni studente deve avere:

- ✓ Il foglio di lavoro: “Kid Fax” (pagina 23)
- ✓ Il foglio di lavoro: “E ora fate i vostri disegni” (page 23)

# Colorare coi numeri

---

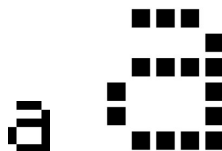
## Introduzione

### Domande di discussione:

1. Come funziona una macchina fax?
2. In quali situazioni i computer memorizzano immagini? (Programmi per il disegno, videogiochi, sistemi multimediali)
3. Come fanno i computer a memorizzare le immagini visto che possono solo usare numeri?

(Si può anche mostrare agli studenti come si spediscono e si ricevono i fax o fare in modo che gli studenti stessi inviino e ricevano fax come preparazione per questa attività)

### Esempio pratico con l'uso di un proiettore di trasparenze.



Gli schermi di un computer sono divisi in una griglia di tanti punti chiamati *pixel* (picture element, elementi di immagine).

In uno schermo in bianco e nero ogni punto può essere o bianco o nero.

La lettera “a” è stata ingrandita qui sopra per mostrare i pixel che la compongono. Quando un computer memorizza una immagine tutto ciò che deve immagazzinare è l'informazione di quali punti debbano essere bianchi e quali neri.

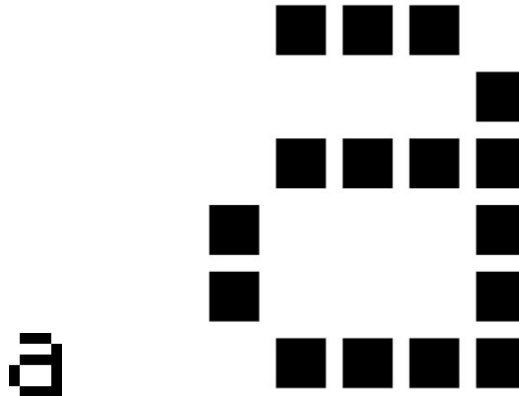
	■	■	■		1, 3, 1
				■	4, 1
	■	■	■	■	1, 4
■				■	0, 1, 3, 1
■				■	0, 1, 3, 1
	■	■	■	■	1, 4

La figura qui sopra mostra come una immagine possa essere rappresentata tramite numeri. La prima riga è composta da un pixel bianco, tre neri e uno bianco. Viene quindi rappresentata come 1, 3, 1.

Il primo numero è sempre relativo al numero dei pixel bianchi all'inizio della linea. Se il primo pixel è nero la linea inizierà con uno 0.

Il foglio di lavoro di pagina 22 contiene alcune immagini che gli studenti possono decodificare usando il metodo appena mostrato.

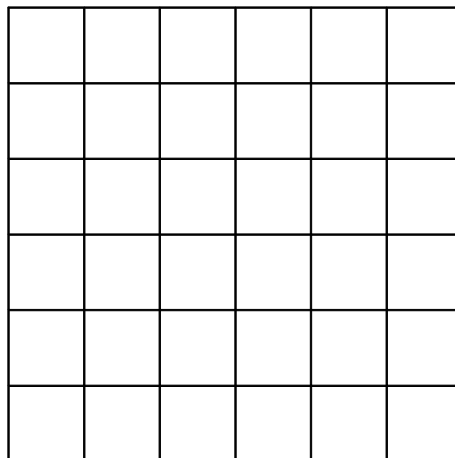
# Lucido da fotocopiare: colorare coi numeri



È la lettera "a" come appare sullo schermo di un computer e ingrandita per mostrare i pixel che la compongono

	■	■	■		1, 3, 1
				■	4, 1
	■	■	■	■	1, 4
■				■	0, 1, 3, 1
■				■	0, 1, 3, 1
	■	■	■	■	1, 4

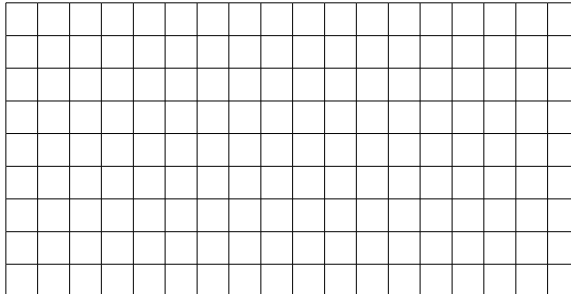
La stessa immagine codificata mediante numeri



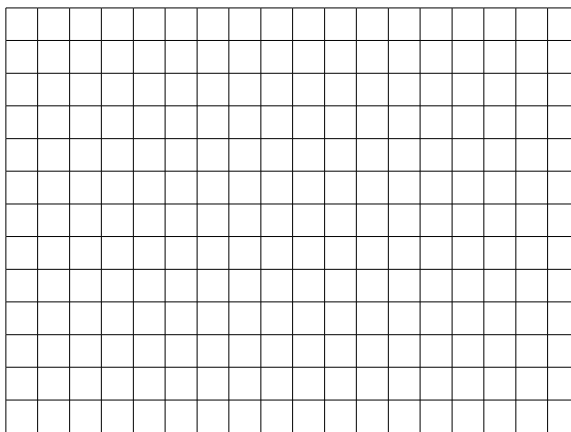
Griglia vuota per scopi didattici

# Foglio di lavoro: Kid Fax

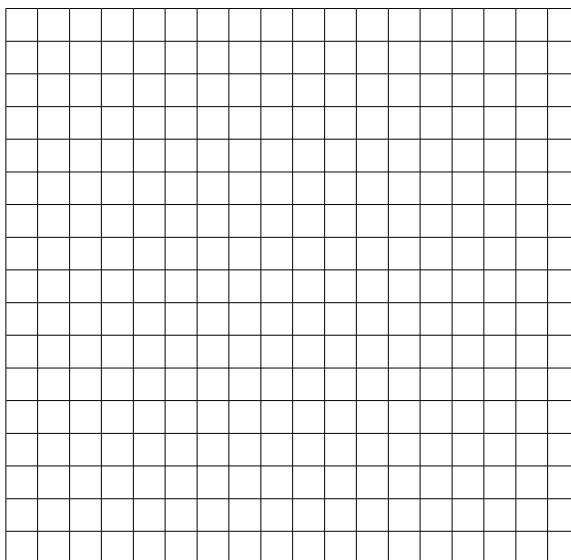
La prima figura è la più facile e l'ultima è la più complessa. E' facile fare errori e quindi vi consigliamo di usare una matita colorata e di avere una gomma a portata di mano!



4, 11  
 4, 9, 2, 1  
 4, 9, 2, 1  
 4, 11  
 4, 9  
 4, 9  
 5, 7  
 0, 17  
 1, 15



6, 5, 2, 3  
 4, 2, 5, 2, 3, 1  
 3, 1, 9, 1, 2, 1  
 3, 1, 9, 1, 1, 1  
 2, 1, 11, 1  
 2, 1, 10, 2  
 2, 1, 9, 1, 1, 1  
 2, 1, 8, 1, 2, 1  
 2, 1, 7, 1, 3, 1  
 1, 1, 1, 1, 4, 2, 3, 1  
 0, 1, 2, 1, 2, 2, 5, 1  
 0, 1, 3, 2, 5, 2  
 1, 3, 2, 5



6, 2, 2, 2  
 5, 1, 2, 2, 2, 1  
 6, 6  
 4, 2, 6, 2  
 3, 1, 10, 1  
 2, 1, 12, 1  
 2, 1, 3, 1, 4, 1, 3, 1  
 1, 2, 12, 2  
 0, 1, 16, 1  
 0, 1, 6, 1, 2, 1, 6, 1  
 0, 1, 7, 2, 7, 1  
 1, 1, 14, 1  
 2, 1, 12, 1  
 2, 1, 5, 2, 5, 1  
 3, 1, 10, 1  
 4, 2, 6, 2  
 6, 6







### **Variazioni e estensioni.**

1. Provate a far colorare con un foglio di carta da lucido posto sulla griglia, così l'immagine finale non avrà la griglia e sarà quindi più chiara.
2. Invece che colorare gli elementi della griglia su di un foglio i ragazzi possono attaccare quadrati di carta adesiva o porre oggetti colorati in una griglia più grande.

### **Punti di discussione.**

C'è di solito un limite alla lunghezza massima della sequenza di pixel dello stesso colore data dal massimo numero binario rappresentabile. Come rappresentereste una sequenza di dodici pixel neri se il vostro sistema usa solo tre bit per ogni sequenza e quindi il numero massimo è sette? (Un buon metodo è di scrivere la sequenza come una sequenza di sette pixel neri, zero pixel bianchi e quindi cinque neri).

---

# Cosa c'entra tutto questo?

---

Una macchina per il fax è un semplice computer che scandisce una pagina in bianco e nero e la trasforma in una griglia di circa 1000x2000 pixel, trasforma le righe in numeri come abbiamo visto, quindi trasferisce i numeri a una macchina simile usando un modem che la ritrasforma in pixel e quindi la stampa su un foglio. Spesso i fogli inviati contengono lunghe sequenze di pixel bianchi (per esempio i margini) o di pixel neri (una linea orizzontale).

Anche le immagini a colori hanno molti pixel ripetuti. Per risparmiare sullo spazio di memorizzazione necessario per mantenere queste immagini i programmatori hanno sviluppato una serie di altri metodi di compressione. Il metodo descritto in questa attività è conosciuto in letterature come “run-length encoding” (codifica di lunghezza delle sequenze), ed è un metodo molto efficace per comprimere le immagini. Se non avessimo compresso le immagini sarebbe stato necessario molto più tempo per trasmettere le immagini e molto più spazio per memorizzarle. Le immagini non compresse sarebbero poco pratiche per i fax o per mettere fotografie in una pagina web. Per esempio tipicamente i fax trasmettono un settimo dei dati che sarebbero necessari per trasmettere le immagini pixel per pixel. Senza la compressione occorrerebbe sette volte più tempo per trasmettere o ricevere un fax.

Le fotografie e le immagini sono di solito compresse con tecniche che consentono di usare da un decimo a un centesimo dei dati necessari per l'immagine originale (usando altre tecniche). Questo consente a molte immagini di poter venir memorizzate su di un disco e occorre anche una quantità molto minore di tempo per poterle recuperare e vedere.

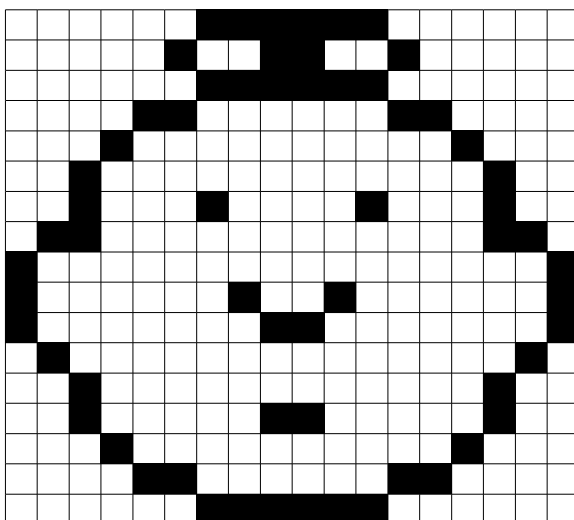
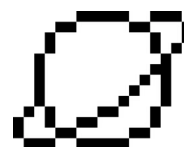
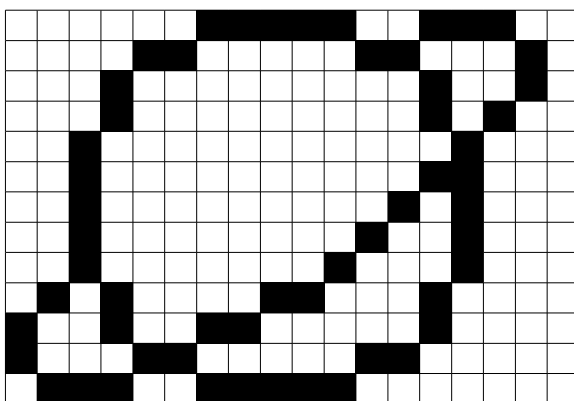
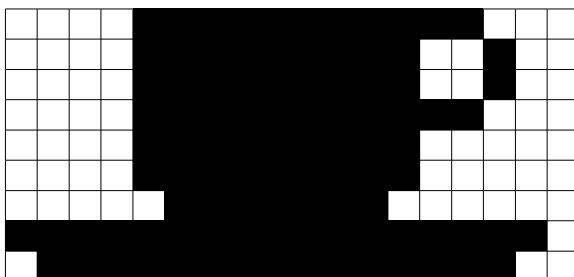
Un programmatore può scegliere fra varie tecniche di compressione per trovare quella più adatta alle immagini da trasmettere o da memorizzare.



# Soluzioni e suggerimenti

---

Risposte al Foglio di lavoro: Kid Fax



# Attività 3

---

## Puoi dirlo nuovamente! –*La compressione del testo*

### Sommario

Dato che i computer hanno solo uno spazio limitato per poter immagazzinare le informazioni devono rappresentarle nel modo più efficiente possibile. Si usano quindi metodi di compressione dell'informazione. Comprimendo i dati per memorizzarli e decomprimendoli per rileggerli, si possono conservare più informazioni o spedirle più velocemente attraverso Internet.

### Competenze richieste

- ✓ Copiare testo scritto

### Età

- ✓ A partire da 9 anni

### Materiale

- ✓ trasparenze da proiettore ottenute fotocopiando il prototipo “puoi dirlo nuovamente!” (pagina 30)

Ogni studente deve avere:

- ✓ Il foglio di lavoro: puoi dirlo nuovamente! (pagina 32)
  - ✓ Il foglio di lavoro: extra per esperti (pagina 33)
  - ✓ Il foglio di lavoro: Apelle, figlio di Apollo (pagina 33)
  - ✓ Il foglio di lavoro: extra per veri esperti (pagina 35)
-

# Puoi dirlo nuovamente!

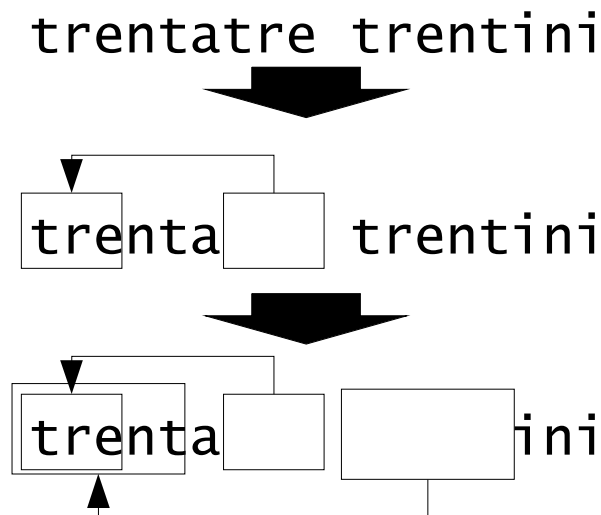
---

## Introduzione

I computer devono memorizzare e trasmettere tantissimi dati. Quindi devono evitare di usare troppo spazio o impiegare troppo tempo per trasmetterli lungo le reti. Questa attività mostra come si possono comprimere i testi.

## Dimostrazione e discussione.

Mostrate il lucido di pagina 30. Osservate quali sequenze di lettere che sono ripetute nel testo. Potete trovare i gruppi di due o più lettere che sono ripetute, o anche intere parole o gruppi di parole. Sostituite le sequenze ripetute tracciando un rettangolo attorno come nel diagramma che segue.



Lucido da fotocopiare: puoi dirlo nuovamente!

---

trentatre trentini  
entrarono a Trento  
tutti e trentatre  
trotterellando

---

## Foglio di lavoro: puoi dirlo nuovamente!

Molte delle lettere e delle parole della poesia che segue sono mancanti. Potete riempire le lettere mancanti e le parole per completare correttamente il testo? Troverete le lettere e le parole mancanti nel riquadro puntato dalla freccia



Ora potete scegliere altre semplici poesie, ninna nanne e creare un simile puzzle. Controllate che le frecce puntino sempre ad una parte precedente del testo. La vostra poesia deve sempre essere ricomposta scorrendo il testo da sinistra a destra, dall'alto al basso, nell'ordine solito della lettura.

**Sfida:** Vedere quante poche parole e lettere della poesia originale occorre tenere!

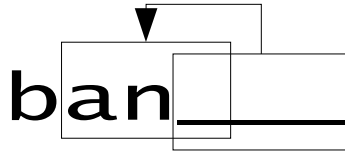
Altri suggerimenti: usate scioglilingua (sopra la panca..., se il coniglio gli agli piglia..., ...) oppure poesie o parole di canzoni con ripetizioni.

**Suggerimento:** Tentate di evitare un sovraffollamento di frecce. Lasciate molto spazio fra le lettere e fra le parole così che ci sia spazio di disegnare i riquadri e le frecce che puntano ad essi.

È più facile disegnare il puzzle se prima scrivete la poesia e poi decidete quali debbano essere i riquadri da tracciare.

## Foglio di lavoro: extra per esperti

Come risolvereste questo puzzle?



Talvolta le parti mancanti del testi puntano a una parte di se stesse. In questo caso occorre copiare lettera per lettera da sinistra a destra per poter ritrovare il testo originario. In questo modo ogni lettera è già disponibile quando è necessaria per proseguire. Questo metodo è utile quando i computer si trovano lunghe sequenze di un particolare carattere o di un gruppo di caratteri.

Potete provare a creare altri esempi per conto vostro.

I computer rappresentano i riquadri e le frecce con numeri, per esempio:

### Banana

può essere scritto come **Ban(2,3)**. "2" significa che occorre tornare indietro di due caratteri per iniziare la copia.

**Ban---**

e "3" significa che occorre copiare tre caratteri consecutivi.

**Bana--**

**Banan-**

**Banana**



cosa stai facendo?

sto comprimendo le banane

Siccome vengono utilizzati due numeri per indicare che ci sono caratteri ripetuti, solo gruppi di due o più lettere vengono compressi, altrimenti non c'è alcun risparmio di spazio. Infatti la lunghezza di un testo si allungherebbe se utilizzassimo due numeri al posto di una lettera.

Scrivete alcune parole nel modo usato da un computer per comprimerle, cercate le parole più interessanti per la compressione. Riescono i vostri amici a ricostruire le parole che avete compresso?



## Foglio di lavoro: Apelle, figlio di Apollo

### Quante parole sono veramente necessarie qui?

Pensate di essere un computer che voglia mettere sul proprio disco più informazioni che si può. Trovate tutte le sequenze di due o più lettere che sono già comparse prima nel testo. Queste sequenze non sono più utili e possono essere sostituite con un puntatore (un riquadro vuoto e una freccia che indica un altro riquadro con le lettere da copiare). Il vostro scopo è di trovare più sequenze che potete.

Apelle figlio di Apollo

Fece una palla di pelle di pollo

Tutti i pesci vennero a galla

Per vedere la palla di pelle di pollo

Fatta da Apelle figlio di Apollo

## Foglio di lavoro: *extra per veri esperti*

### **Siete pronti per una compressione veramente *dura*?**

La favola che segue è stata analizzata da un programma per computer (uno di quelli veri con la spina) che ha trovato che ci sono almeno 499 sequenze e 2281 lettere che possono essere eliminate perché appartengono a sequenze ripetute (la divisione fra righe e paragrafi viene considerata come uno spazio). Quante sequenze riuscite a trovare? Quanti caratteri in totale potete eliminare? Ricordatevi che solo le sequenze di due o più caratteri ripetuti possono essere eliminate. Buona Fortuna!

**C**'erano una volta, tanto tanto tempo fa, tre porcellini che volevano andare a vivere da soli. Il primo porcellino non era molto furbo e decise di costruire la sua casa con la paglia perché costava poco. Anche il secondo porcellino non era furbo e costruì la sua casa di legno perché aveva un look ecologico ed era di moda. Il terzo porcellino era più furbo dei suoi due fratelli e comperò un carico di mattoni in una città vicina coi quali costruì una casa modesta ma confortevole.

Poco tempo dopo la festa di inaugurazione della casa, il primo porcellino era assorto su una sedia leggendo un libro quando udì bussare alla porta. Era il lupo cattivo, naturalmente.

"Porcellino, porcellino, lasciami entrare!" gridò il lupo.

"Neanche per un pelo della mia cotenna!" rispose il primo porcellino.

"Allora sbufferò, soffierò e abatterò la tua casa" ruggì il lupo, e sbuffò, soffiò e la casa in breve crollò. Il primo porcellino corse più forte che poté verso la casa di legno, e presto fu al sicuro lì col fratello. Ma non passò molto che il lupo arrivò anche lì.

"Porcellino, porcellino, lasciami entrare!" gridò il lupo.

"Neanche per un pelo della mia cotenna!" rispose il secondo porcellino.

"Allora sbufferò, soffierò e abatterò la tua casa" ruggì il lupo, e sbuffò, soffiò e presto della casa non rimase che un mucchio di legna per il camino. Il due porcellini terrorizzati corsero più forte che poterono verso la casa di mattoni, ma il lupo era sempre alle loro calcagna e appena i due porcellini arrivarono alla casa del loro fratello, anche il lupo arrivò ancora una volta sulla soglia.

"Porcellino, porcellino, lasciami entrare!" gridò il lupo.

"Neanche per un pelo della mia cotenna!" rispose il terzo porcellino.

"Allora sbufferò, soffierò e abatterò la tua casa" ruggì il lupo, e sbuffò, soffiò e, naturalmente, la casa era fatta di mattoni e il lupo rimase senza fiato. Allora ebbe un'idea: il camino! Si arrampicò su una quercia e salì sul tetto. Ma il terzo porcellino che era molto sensibile ai temi ambientali invece del camino aveva installato pannelli solari. Frustrato il lupo scivolò e cadde dal tetto rompendosi la zampa sinistra, danneggiando irreparabilmente la propria autostima. Quando il lupo se ne andò i porcellini risero e pensarono a quanto sicuro sia vivere in città dove i lupi si possono trovare solo allo zoo. E così fecero e, naturalmente, vissero felici e contenti.

# Cosa c'entra tutto questo?

---

La capacità di memorizzazione dei computer sta crescendo ad un ritmo incredibile, negli ultimi 25 anni la quantità di dati immagazzinabili è crescita di circa un milione di volte, ma continuiamo a trovare informazioni da mettere nei computer. I computer possono contenere interi libri o intere biblioteche, e ora musica e film a patto che abbiano spazio a sufficienza. I grandi archivi (file) sono anche un problema sulla rete Internet perché hanno necessità di lunghi tempi per poter essere trasferiti. I computer stanno anche diventando sempre più piccoli, anche un telefono cellulare o un orologio da polso contengono moltissime informazioni.

C'è comunque una soluzione per risolvere questo problema. Invece che acquistare sempre più spazio per immagazzinare dati, per esempio dischi fissi o removibili, oppure invece che acquistare modem più veloci e comperare linee a velocità più elevata possiamo *comprimere* i dati. Il processo di compressione e di decompressione dei dati è effettuato automaticamente dal computer. Tutto quello che possiamo notare è che il disco contiene più dati e che le pagine web vengono trasferite più velocemente, ma il computer deve fare più calcoli.

Sono stati inventati molti metodi di compressione per i testi. Il metodo usato in questa attività, sfruttando il principio di riferirsi alla precedente occorrenza di sequenze ripetute, è noto con il nome di “codifica di Ziv-Lempel” o più semplicemente come codifica “ZL” perché è stata inventata da due professori israeliani (il prof. Ziv e il prof. Lempel) negli anni '70. È qualche volta indicata anche come “zip” nei personal computer e viene anche usata nelle immagini GIF e nei modem ad alta velocità. Nel caso dei modem, riduce la quantità di dati che devono essere scambiati sulla linea telefonica e quindi il collegamento risulta più veloce.

Alcuni altri metodi sono basati sull'idea che alcune lettere sono più frequenti di altre e quindi vengono associati codici più corti alle lettere più frequenti. Il codice Morse usa questa idea.

## Soluzioni e suggerimenti

---

**Puoi dirlo nuovamente!** (pagina 32)

**stella stellina,  
la notte si avvicina  
la fiamma traballa  
la mucca è nella stalla  
la mucca col vitello  
la pecora e l'agnello  
la chioccia col pulcino  
ognuno ha il suo bambino  
ognuno ha la sua mamma  
e tutti fan la nanna.**

# Attività 4

---

## La magia delle carte girate—*Il riconoscimento e la correzione degli errori*

### Sommario

Quando i dati vengono memorizzati su un disco o trasmessi da un computer all'altro, noi assumiamo che non vengano modificati. Purtroppo talvolta le cose non vanno così e i dati vengono accidentalmente cambiati. Questa attività mostra un trucco magico per poter scoprire quali dati sono stati rovinati e come poterli correggere.

### Competenze richieste

- ✓ Saper contare
- ✓ Riconoscere i numeri pari e i numeri dispari

### Età

- ✓ A partire da 9 anni

### Materiale

- ✓ 36 fermagli magnetici “da frigo” colorati su di un lato.
  - ✓ Una superficie dove attaccare i magneti (di solito le lavagne bianche vanno bene).
- Ogni coppia di studenti deve avere.
- ✓ 36 carte identiche colorate da un solo lato.
-

# Il “trucco magico”

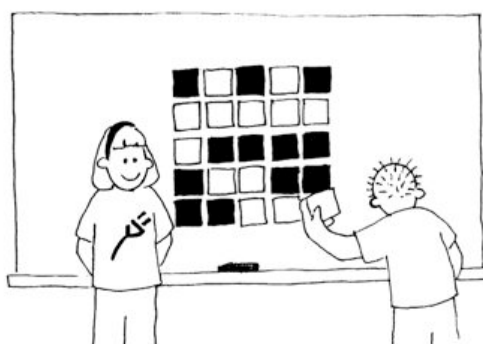
---

## Dimostrazione

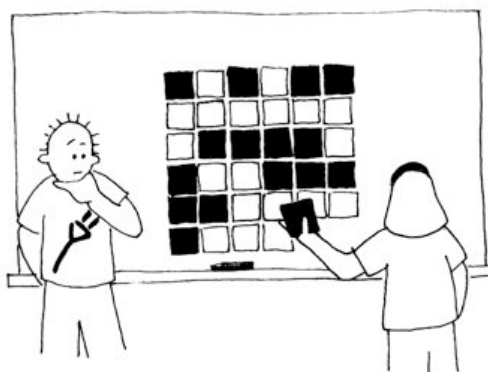
Questa è la vostra possibilità per essere un mago.

Avete bisogno di una pila di carte identiche colorate in modo diverso sui due lati (potete costruirle ritagliando un cartone colorato da un lato solo). Per mostrare il metodo agli studenti è più semplice usare calamite che abbiamo colori diversi sui due lati, esistono perfette calamite “da frigo” con queste caratteristiche.

1. Scegliete uno studente/una studentessa e chiedetegli/le di disporre le carte a formare un quadrato 5 x 5, scegliendo a caso i lati delle carte visibili.



Aggiungete voi una riga e una colonna, “giusto per fare le cose un po' più difficili”.



Queste carte sono la chiave del trucco. Non le metterete a caso ma le disporrete in modo che tutte le righe e le colonne abbiano un numero pari di carte colorate.

2. Chiedete ora ad un altro studente o a un'altra studentessa di girare una sola delle carte mentre voi vi coprite gli occhi. La riga e la colonna dove è stata girata la carta contengono un numero dispari di carte colorate. In questo modo potete individuare la carta cambiata.

Ora gli studenti sono capaci di indovinare il trucco?

### **Insegnate il trucco agli studenti:**

1. Lavorando in coppie gli studenti dispongono le carte a formare un quadrato 5x5.
2. Quante carte colorate ci sono in ogni riga e in ogni colonna? È un numero pari o dispari? Ricordate che zero è un numero pari.
3. Ora aggiungete una carta in ogni riga in modo che il numero di carte colorate di ogni riga sia pari. Questa carta aggiuntiva viene chiamata la “parità”.
4. Aggiungete una riga in fondo in modo che ogni colonna contenga un numero pari di carte colorate.
5. Ora girate una carta. Come potete trovare la riga e la colonna della carta girata? (avranno un numero dispari di carte colorate). I bit di parità servono proprio a segnalare quando viene commesso un errore.
6. Ora fate a turni per provare il “trucco”.

### **Proposte di estensione:**

1. Provate a usare altri oggetti. Tutto ciò che ha due “stati” va bene. Per esempio si possono usare carte da gioco (rosso o nero), monete (testa o croce) o carte con 0 da un lato e 1 dall'altro (per correlare l'attività ai numeri binari).
  2. Cosa accade se due o più carte vengono girate? (non è sempre possibile dire che quali due carte siano state cambiate. In generale si possono individuare due coppie di carte fra le quali ci sono quelle modificate. Cambiando quattro carte è possibile che i bit di parità siano corretti anche dopo la modifica e quindi l'errore rimarrebbe non riconosciuto).
  3. Un altro esercizio interessante consiste nell'analizzare l'ultima carta in basso a destra. Per come è stata descritto il metodo questa carta viene calcolata per avere un numero pari di carte colorate nella colonna delle carte di parità di ogni riga. È anche essa stessa la carta di parità per la ultima riga? (Sì sempre).
  4. In questo esercizio abbiamo usato la parità (il numero di carte colorate in ogni riga e colonna deve essere pari). È se invece usassimo la disparità (il numero delle carte colorate nelle righe e nelle colonne deve essere dispari). Si può fare lo stesso ragionamento del punto precedente? l'ultima carta in basso a destra ricostruisce la disparità sia dell'ultima riga sia dell'ultima colonna? (Può succedere ma in generale funziona se il numero di righe e di colonne sono uguali o almeno sono entrambe pari o entrambe dispari. Funziona per il caso 5x5 ma anche per 5x9 o 4x6 ma non per 3x4).
-

## Un esempio per esperti, tratto dalla vita reale.

La stessa tecnica è usata con i codici identificativi dei libri. Tutti i testi pubblicati hanno un codice numerico di identificazione formato da 10 cifre, di solito stampato nell'ultima di copertina. La decima cifra è una cifra di controllo, esattamente come il bit di parità nell'esercizio.

Questo significa che se ordinate un libro usando il codice ISBN (International Standard Book Number), l'editore può controllare che non abbiate commesso errori di trascrizione controllando la cifra di controllo. Questo fa in modo che l'errore possa essere scoperto prima che vi troviate con un libro diverso da quello che volevate ordinare.

Ecco come funziona il meccanismo di controllo:

Moltiplicate la prima cifra per dieci, la seconda per nove, la terza per 8 e così via fino alla nona cifra che va moltiplicata per due e sommate tutti questi prodotti insieme.

Per esempio per il codice ISBN 0-13-911991-4 si ottiene il seguente valore:

$$\begin{aligned} & (0 \times 10) + (1 \times 9) + (3 \times 8) + (9 \times 7) + (1 \times 6) \\ & + (1 \times 5) + (9 \times 4) + (9 \times 3) + (1 \times 2) \\ & = 172 \end{aligned}$$

Dividete ora il risultato per 11. Quale è il resto di questa divisione?

$$172 \div 11 = 15 \text{ resto } 7$$

Se il resto è zero, allora la cifra di controllo (detta checksum in inglese) è zero, altrimenti sottraete da 11 il resto per trovare il valore della cifra di controllo.

$$11 - 7 = 4$$

Ora potete controllare. È l'ultima cifra del codice ISBN? Sì! Allora funziona davvero!

Se l'ultima cifra fosse stata diversa da quattro, allora avremmo scoperto che il codice ISBN era errato, magari una cifra era stata copiata male.

È possibile che la cifra di controllo risulti 10 e quindi non basterebbe un solo numero per rappresentarla. In questo caso viene usato il carattere X.



## Un esempio di codice a barre di un prodotto alimentare

Un altro caso nel quale viene usata una cifra di controllo sono i codici a barre che trovate sui prodotti che comperate nei negozi e nei supermercati. Il concetto è lo stesso ma in questo caso si usa una diversa formula. Nei supermercati le casse hanno uno scanner: uno strumento che proiettando dei raggi laser, di solito di colore rosso, legge il codice a barre per riconoscere il prodotto acquistato e trovare il relativo prezzo. Se alla cassa lo scanner non legge bene il codice, la cifra di controllo non risulterà corretta e la cassa non farà il caratteristico bip. L'operatore tenterà di passare l'articolo nuovamente sotto lo scanner e alla fine se lo scanner non riesce a leggere il codice scriverà i numeri su di una tastiera, codice di controllo compreso. Così si controlla anche l'operatore che abbia scritto i numeri corretti!



# Controllate quel libro

Agenzia Investigativa

Controllalibri



Cerchiamo e controlliamo i codici ISBN. Prezzi Modici.

Unitevi alla nostra agenzia, cercate nei libri in classe o nella biblioteca i codici ISBN.

## Le cifre di controllo sono corrette?

Talvolta vengono commessi errori nella trascrizione dei codici ISBN.

Alcuni degli errori comuni sono:

- ✗ copiare male una cifra;
- ✗ scambiare fra loro due cifre;
- ✗ aggiungere una cifra al numero per esempio copiandola due volte; e
- ✗ non copiare una cifra del numero.

Riuscite a trovare libri che hanno la lettera X al posto della cifra di controllo? Non dovrebbe essere troppo difficile, in media un libro ogni 11 ha la lettera X.

Quale tipo di errori possono capitare senza che vengano riconosciuti dalla cifra di controllo? È possibile cambiare una sola cifra ed avere lo stesso la cifra di controllo corretta? Cosa succede se si scambiano fra loro due cifre (un errore molto comune)?

# Cosa c'entra tutto questo?

---

Immaginate di depositare 10 euro nel vostro conto in banca. L'impiegato scrive l'importo del deposito e il dato viene spedito al computer centrale. Ma supponete ora che ci sia una interferenza lungo la linea e il codice che rappresenta 10 euro venga cambiato in 1000 euro. A voi magari piacerebbe, ma per la banca sarebbe chiaramente un problema.

È importante poter riconoscere gli errori che possono accadere nella trasmissione dei dati. Quindi occorre che il computer ricevente possa controllare che i dati che arrivano non siano stati rovinati da interferenze lungo la linea. Talvolta in caso di errore i dati possono essere ricevuti di nuovo, come nel caso dei dati scambiati tra due computer, il computer mittente può ritrasmetterli. Ci sono situazioni nelle quali non è possibile avere questa seconda possibilità, per esempio quando un disco o un nastro viene rovinato dall'esposizione a campi elettrici o magnetici, al calore o proprio fisicamente rotto. Talvolta anche se i dati possono teoricamente essere ritrasmessi non è conveniente farlo perché ci vorrebbe troppo tempo, i dati trasmessi da una sonda spaziale dal pianeta Giove hanno bisogno di almeno mezz'ora per arrivare fino a noi, sarebbe veramente noioso dover aspettare la ritrasmissione!

Occorrono quindi metodi per riconoscere quando un dato è rovinato (riconoscimento dell'errore o *error detection*) e per ricostruire il dato esatto da quello rovinato (correzione dell'errore o *error recovery*).

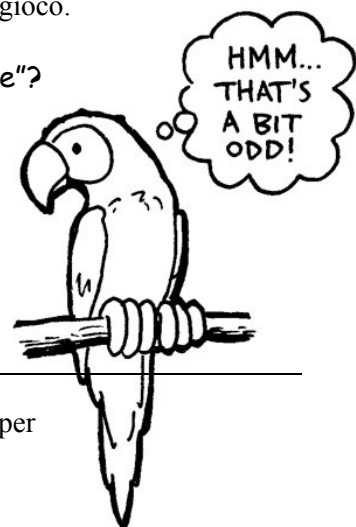
Lo stesso metodo usato nel gioco delle “carte girate” viene usato dai computer. Mettendo i bit in righe e colonne immaginarie non solo possiamo vedere quando accadono errori ma anche dove sono. È possibile quindi ripristinare il valore corretto, e quindi questo è un metodo di correzione dell'errore.

I computer spesso usano anche metodi di controllo più complessi che sono capaci di riconoscere e correggere più errori e non uno solo. I dischi rigidi di un computer hanno vaste aree dedicate alla correzione così che i dischi possano continuare a funzionare anche se parti del disco venissero danneggiate. Questi metodi sono comunque correlati al metodo di controllo della parità che abbiamo studiato.

Siete anche molto bravi con l'inglese? Se sì apprezzerete questo gioco.

**Q:** What do you call this: "Pieces of nine, pieces of nine"?

**A:** A parrot error.



---

## Soluzioni e suggerimenti

Ci sono errori che potrebbero essere non riconosciuti, accadono per esempio quando una cifra viene diminuita e ed un'altra incrementata. La somma potrebbe essere uguale, per esempio se

aggiungete due alla ottava cifra (che viene moltiplicata per tre) e togliete tre alla nona cifra (moltiplicata per due) la somma non cambia.

Ecco la spiegazione del dialogo del pappagallo, se qualche studente (non madrelingua americano/a o inglese) ce l'ha fatta da solo/a a capirne il significato, probabilmente non avrà bisogno di studiare inglese fino all'università!

Traduzione letterale:

Domanda: come chiameresti questo: “pezzi da nove, pezzi da nove”?

Risposta: un errore da pappagallo.

Il pappagallo dice “è un po' strano”

Spiegazione:

Così sembra non avere senso.

Ma il “pezzo da otto” è una antica moneta spagnola in uso anche in Messico. Nel libro “l'isola del tesoro” c'è un pappagallo ammaestrato per gridare a tutti quelli che entrano “pezzi da otto, pezzi da otto”. Nel film “I pirati dei Caraibi” i pirati devono trovare “nove pezzi da otto” e “pezzi da otto” è una nuova attrazione di Disneyland California.

Il pappagallo sbaglia dicendo “pezzi da nove, pezzi da nove”, visto che tale moneta non esiste. Ma errore da pappagallo “parrotty error” suona come “parity error”, errore di parità.

La frase del pappagallo “è un po' strano” visto che le parole “bit” e “odd” hanno due significati può essere letta anche come “è un bit dispari”!

# Attività 5

---

## Indovina indovinello – *La Teoria dell'Informazione*

### Sommario

Quanta informazione c'è in un libro di 1000 pagine? C'è più informazione in un elenco telefonico di 1000 pagine oppure in una risma di carta di 1000 pagine bianche, oppure nel libro di Tolkien *Il Signore degli Anelli*? Se riuscissimo a misurarla, allora potremmo stimare quanto spazio occorre per memorizzarne una data quantità. Per esempio, riesci a leggere quello che segue?

L frs h l vcl mncnt

Probabilmente sì, perché non c'è 'troppa informazione' nelle vocali. Questa attività presenta un modo per misurare il contenuto di informazione.

### Abilità

- ✓ Confronto di numeri e operazioni con intervalli numerici
- ✓ Deduzioni logiche
- ✓ Capacità di fare domande

### Età

- ✓ A partire da 10 anni

### Materiali

- ✓ Nessun materiale per la prima attività

Per l'attività estesa, ciascun bambino ha bisogno di:

- ✓ Foglio di lavoro: alberi di decisione (*pagina 49*)
-

# Indovina indovinello

---

## Discussione

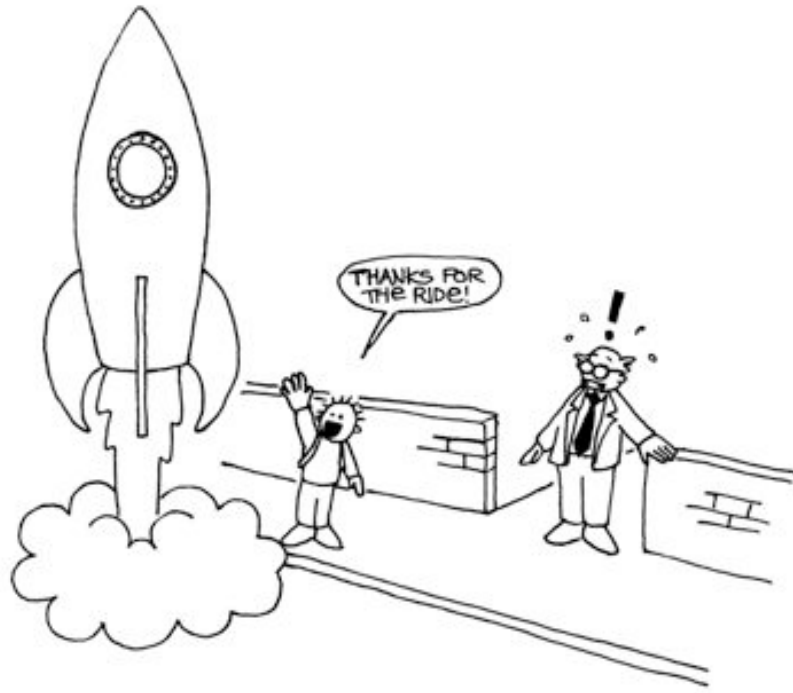
1. Discutete con i bambini cosa pensano che sia l'informazione.
2. Come potremmo misurare l'informazione che c'è in un libro? Ha a che vedere con il numero di pagine oppure sono importanti le parole? Un libro può avere più informazione di un altro? E se un libro fosse noioso e un altro particolarmente interessante? Un libro di 400 pagine che contenesse la frase “bla, bla, bla” avrebbe più o meno informazione di un elenco telefonico?

Basta spiegare che gli informatici misurano l'informazione considerando quanto è 'sorprendente' un libro o un messaggio! Considerando qualcosa che già conoscete – per esempio, quando un compagno che viene sempre a scuola a piedi dice “Io sono venuto a scuola a piedi oggi” — non dà nessuna informazione per il semplice fatto che non è sorprendente. Se il vostro compagno dicesse invece “oggi ho avuto un passaggio fino a scuola con un elicottero”, quello sì che *sarebbe* sorprendente, e fornirebbe un sacco di informazione.

Come può essere misurato il valore di sorpresa di un messaggio?

Un modo è verificare quanto sia difficile indovinare l'informazione. Se il compagno dicesse, “Indovina come sono venuto a scuola oggi” e fosse venuto a piedi, probabilmente potreste indovinarlo la prima volta. Ci potrebbero volere un po' più tentativi per arrivare all'elicottero e ancor di più se il compagno fosse venuto con una navicella spaziale.

La quantità di informazione che il messaggio contiene è misurata attraverso quanto sia facile o difficile indovinarla. Il gioco che segue ce ne dà un'idea.



“Grazie per il passaggio!”

# Attività dell'indovina indovinello

---

Prima si sceglie un bambino. Gli altri bambini possono rivolgergli domande, ma questi potrà rispondere solo sì o no fino a che la risposta sia stata indovinata. Qualsiasi domanda è pertinente, purché la risposta sia strettamente ‘sì’ o ‘no’.

## Suggerimenti:

State pensando a:

- ✓ un numero tra 1 e 100
- ✓ un numero tra 1 e 1000
- ✓ un numero tra 1 e 1,000,000.
- ✓ un qualsiasi numero intero
- ✓ una sequenza di 6 numeri con una logica, da indovinare in ordine dal primo all'ultimo (ad esempio 2, 4, 6, 8, 10)

Contate il numero di domande fatte. Questa è una misura del valore dell'“informazione”.

## Discussione che segue

Quale strategia avete usato? Quale era la migliore?

Considerate che solo 7 domande possono essere poste per indovinare un numero tra 1 e 100 se dimezzi l'intervallo ogni volta. Per esempio:

È inferiore a 50?	Sì.
È inferiore a 25?	No.
È inferiore a 37?	No.
È inferiore a 43?	Sì.
È inferiore a 40?	No.
È inferiore a 41?	No.
Dev'essere 42!	Sì!

Sorprendentemente, se l'intervallo numerico fosse incrementato fino a 1000, non ci sarebbe uno sforzo 10 volte maggiore, — sarebbero sufficienti solo 3 o 4 domande in più. Ogni volta che l'intervallo raddoppia si ha necessità di una domanda in più per trovare la risposta.

Un buon modo per approfondire l'argomento è quello di lasciare i bambini giocare a Mastermind.

## Estensione: Quanta informazione c'è in un messaggio?

Gli informatici non hanno a che fare solo con numeri ma anche con le lettere e possono quindi indovinare quale lettera è più probabile in una parola o in una frase.

Provate a giocare con un frase corta di 4–6 parole. Le lettere devono essere indovinate nell'ordine corretto, dalla prima all'ultima. Occorre qualcuno che scriva le lettere man mano che sono trovate e tenga traccia di quanti tentativi occorrono per trovare ciascuna lettera. Qualsiasi domanda con sì/no può essere considerata. Un esempio potrebbe essere, “è una *t*?” “è una vocale?” “viene prima di *m* nell'alfabeto?” Lo spazio tra le lettere è esso

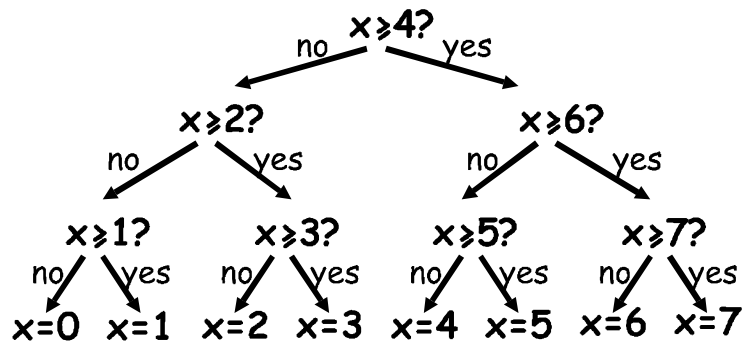
stesso una lettera e può essere indovinata. Provate a verificare quali parti del messaggio sono più facili da indovinare.



## Foglio di lavoro: alberi di decisione

Se già conoscete la strategia per fare domande, allora potete trasmettere un messaggio senza chiedere nulla.

Ecco qui uno schema chiamato albero di decisione (decision tree) per indovinare i numeri tra 0 e 7:



Quante sono le decisioni yes/no (sì/no) per indovinare il numero 5?

Quante sono quelle necessarie per indovinare un numero qualsiasi?

Approfondite ora una caratteristica molto interessante. Sotto i numeri 0, 1, 2, 3... nella riga finale dell'albero scrivete il numero in binario (Attività 1).

Guardate attentamente l'albero. Se no=0 e yes/sì=1 cosa ne ricavate?

Nel gioco del numero da indovinare cercate di scegliere domande tali che la sequenza di risposte che ne verrà rappresenti il numero esattamente in questo modo.

Scrivete il vostro albero di decisione per indovinare numeri tra 0 e 15.

**Extra per gli esperti:** Quale tipo di albero usereste per indovinare l'età di qualcuno?

E quale per indovinare la successiva lettera in una frase?

# Cosa c'entra tutto questo?

---

Claude Shannon, che era un matematico americano molto famoso (oltre ad essere matematico era un grande saltimbanco e un ottimo mono-ciclista), fece molti esperimenti con questo gioco. Egli misurò la quantità di informazione in 'bits': ogni risposta sì/no è equivalente al bit 1/0. Così facendo trovò che la quantità di "informazione" contenuta in un messaggio dipende da ciò che già si sa. Qualche volta possiamo fare una domanda che elimina la necessità di porne molte altre. In questo caso, il contenuto di informazione del messaggio è basso. Per esempio, se si lancia una volta una moneta l'informazione è normalmente di 1 bit: testa o croce. Ma se la moneta fosse truccata e testa uscisse 9 volte su dieci, allora l'informazione non sarebbe più 1 bit: ci si creda o no, sarebbe inferiore. Come si fa a sapere quale è il risultato del lancio della moneta con *meno* di una domanda sì/no? Semplice, basta usare domande come "i prossimi *due* lanci danno entrambi testa?" Per una sequenza di lanci con la moneta truccata, la risposta sarà "sì" circa l'80% delle volte. Sul 20% delle volte dove la risposta è "no," dovrete porvi altre due domande per trovare il caso esatto. Ma in media, avrete posto meno di una domanda per lancio di moneta!



Shannon chiamò con il termine "entropia" il contenuto informativo di un messaggio. L'entropia dipende non solo dal *numero* dei possibili esiti—nel caso di una moneta, due—ma anche dalla *probabilità* che ciò accada. Eventi improbabili, o informazioni infrequenti, hanno bisogno di molte più domande per indovinare il messaggio perché questi portano molta informazione che non si conosceva—proprio come prendere un elicottero per andare a scuola.

L'entropia di un messaggio è molto importante per gli informatici. Non si può infatti comprimere un messaggio al di sotto della sua entropia in bit e il miglior sistema di compressione è equivalente al gioco delle domande. Poiché è un programma che pone le 'domande', la sequenza può essere riprodotta in un secondo momento e se si memorizzano le risposte (in bit) l'informazione può essere ricostruita! Il miglior sistema di compressione può ridurre file di testo a circa un quarto della dimensione originaria—un grosso risparmio di spazio!

Il metodo degli indovinelli può anche essere usato per costruire una interfaccia per computer che predica cosa l'utente digiterà! Questo è molto utile per persone con disabilità fisiche che trovano difficile digitare con la tastiera. Il computer suggerisce cosa probabilmente sarà digitato e l'utente sceglie la lettera corretta. Un buon sistema ha bisogno solo di due risposte sì/no per ciascun carattere e può essere di grande aiuto per coloro che hanno difficoltà nel compiere i fini movimenti che controllano il mouse o la tastiera. Un sistema analogo è usato anche per digitare un testo SMS nei cellulari.

---

# Soluzioni e suggerimenti

---

La risposta ad una domanda sì/no corrisponde esattamente ad un bit di informazione—sia con una domanda semplice “è più di 50?” o complessa “è tra 20 e 60?”

Nel gioco del numero da indovinare, le domande sono scelte in un certo modo: la sequenza delle risposte è proprio la rappresentazione binaria del numero. Tre è 011 in binario ed è rappresentato dalle risposte “No, sì, sì” nell'albero di decisione e, come visto, possiamo scrivere no al posto di 0 e sì al posto di 1 e viceversa.

Un albero di decisione per l'età potrebbe avere a che vedere con numeri ancora inferiori di domande, poiché l'aspetto della persona è già un'informazione che abbiamo e che influenza il resto.

L'albero di decisione relativo alle lettere successive in una sequenza dipende sicuramente dalle lettere precedenti.

---

# Parte II

*Far lavorare i computer—  
Algoritmi*

# Far lavorare i computer

---

I computer agiscono seguendo una sequenza di istruzioni. Queste istruzioni gli permettono, ad esempio, di ordinare, trovare e mandare informazioni. Per fare queste cose nel minor tempo possibile si ha la necessità di metodi efficienti sia per trovare quel che si cerca in grosse moli di dati sia per comunicare attraverso la rete.

Un *algoritmo* è un insieme di istruzioni preparato per completare un lavoro. L'idea di algoritmo è centrale per l'informatica. Gli algoritmi, infatti, sono come noi diciamo ai computer di risolvere i problemi. Alcuni algoritmi sono più veloci di altri e la maggior parte degli algoritmi che sono stati scoperti hanno reso possibile risolvere problemi che prima richiedevano una incredibile quantità di tempo—per esempio, scoprire milioni di cifre del pi greco o tutte le pagine che contengono un dato nome nel World-Wide Web o trovare il miglior modo di stivare pacchi in un container, oppure trovare se numeri molto grandi (100-cifre e più) sono primi.

Il termine “algoritmo” deriva dal nome di Mohammed ibn Musa Al-Khowarizmi— Mohammed, figlio di Moses, da Khowarizm— che operò in un centro accademico noto come la Casa della Sapienza a Baghdad attorno all'800 dopo Cristo. I suoi lavori trasmisero l'arte Hindu del calcolo agli Arabi e in seguito all'Europa. Quando furono tradotti in Latino nel 1120 le prime parole erano “Dixit Algorismi”—“così disse Algorismi”.

# Attività 6

---

## Battaglia navale—*Algoritmi di ricerca*

### Sommario

Ai computer si richiede spesso di cercare informazioni all'interno di grandi moli di dati. Per far ciò, essi necessitano di metodi veloci ed efficienti. L'attività presenta tre diversi tipi di ricerca: lineare, binaria e hash.

### Abilità

- ✓ Ragionamento logico

### Età

- ✓ A partire da 9 anni

### Materiali

Ciascun bambino avrà bisogno:

- ✓ una copia della battaglia navale
  - 1A, 1B per il gioco 1
  - 2A, 2B per il gioco 2
  - 3A, 3B per il gioco 3
- ✓ *si potrebbe aver bisogno anche di qualche copia delle battaglie navali supplementari 1A', 1B', 2A', 2B', 3A', 3B'.*

# Battaglia navale

---

## Attività introduttiva

1. Scegliete 15 bambini da allineare di fronte al resto della classe. Date a ciascun bambino una carta con un numero (in ordine casuale). Mantenete i numeri nascosti al resto della classe.
2. Date ad un altro bambino un contenitore con quattro o cinque caramelle. Il suo scopo è quello di indovinare un dato numero. Il bambino può “pagare” per sbirciare una carta particolare dei suoi compagni in fila. Se indovina il numero corretto prima di usare tutti i dolci che ha, allora può tenere quelli rimasti.
3. Ripetete i passi precedenti con altri bambini.
4. Adesso mescolate le carte e ridatele nuovamente ai bambini. Questa volta, fate in modo che i bambini si mettano in ordine ascendente. Ripetete la ricerca.

Se i numeri sono ordinati, una saggia strategia è quella di effettuare solo un “pagamento” per eliminare metà dei bambini poiché la carta di mezzo di fatto rivela anche i numeri delle carte di una metà della fila. Ripetendo questa strategia, si dovrebbe trovare il numero cercato con solo 3 caramelle. L'incremento di efficienza sarà di conseguenza ovvio.

## Attività

I bambini possono comprendere come un computer effettua una ricerca giocando a battaglia navale. Mentre giocano, fate in modo che pensino alle strategie che stanno usando per localizzare le navi.

---

## Battaglia navale – *Un gioco a ricerca lineare*

---

### **Leggete le istruzioni che seguono ai bambini**

1. Organizzatevi in coppie. Uno di voi ha il foglio 1A, l'altro l'1B. Non mostrate il vostro foglio al compagno!
2. Entrambi cerciate una delle 26 navi della parte alta del foglio e dite al compagno il numero. Quella sarà la vostra nave.
3. Adesso prendetevi tutto il tempo per indovinare dov'è la nave del compagno. (A turni uno dice la lettera della nave (da A a Z) e il compagno comunica il numero della nave che corrisponde a quella lettera nel suo schema. Se il numero corrisponde alla nave scelta dal compagno la nave è *affondata* se no è *acqua*). Usate lo schema nella parte bassa del foglio per barrare le varie navi ad ogni tentativo per evitare di fare richieste inutili (ad esempio chiedendo due volte la stessa lettera).
4. Di quanti colpi avete bisogno per localizzare la nave del compagno? Questo è il vostro punteggio di gioco: vince chi ha meno punti.

(I fogli 1A' e 1B' sono in più per quei bambini che vogliono giocare ancora o che “inavvertitamente” hanno visto i fogli dei compagni. I fogli 2A', 2B' e 3A', 3B' servono per giochi che saranno presentati tra poco).

### **Discussione che segue**

1. Cosa rappresenta il punteggio?
  2. Quale valore potrebbero avere il punteggio minimo e massimo? (Sono 1 e 26 rispettivamente, assumendo che i bambini non sparano alla stessa nave due volte. Questo metodo è chiamato ‘ricerca lineare’ perché si ha la necessità di toccare tutte le posizioni una ad una nel peggiore dei casi).
-



## Battaglia navale – *Un gioco a ricerca binaria*

---

### Istruzioni

Le istruzioni per questa versione del gioco sono le stesse di quello precedente tranne che i numeri delle navi sono in ordine ascendente nei fogli. Spiegate questo ai bambini prima dell'inizio.

1. Organizzatevi in coppie. Uno ha il foglio 2A, l'altro il 2B. **Non** mostrate il vostro foglio al compagno!
2. Entrambi cerciate una delle 26 navi della parte alta del foglio e dite al compagno il numero. Quella sarà la vostra nave.
3. Adesso prendetevi tutto il tempo per indovinare dov'è la nave del compagno. (A turni uno dice la lettera della nave (da A a Z) e il compagno comunica il numero della nave che corrisponde a quella lettera nel suo schema. Se il numero corrisponde alla nave scelta dal compagno la nave è *affondata* se no è *acqua*). Usate lo schema nella parte bassa del foglio per barrare le varie navi ad ogni tentativo per evitare di fare richieste inutili (ad esempio chiedendo due volte la stessa lettera).
4. Di quanti colpi avete bisogno per localizzare la nave del compagno? Questo è il vostro punteggio di gioco: vince chi ha meno punti.

### Discussione che segue

1. Cosa rappresenta il punteggio?
2. Quale strategia hanno usato coloro che hanno fatto meno punti?
3. Quale nave dovrete scegliere per prima? (Quella di mezzo vi dice quale metà scartare e in quale metà è quella da cercare). Quale posizione si dovrebbe scegliere successivamente? (Di nuovo, la miglior strategia è quella di scegliere la nave nel mezzo della parte dove si sa dove è quella da scovare).
4. Se si applica questa strategia, quanti colpi sono necessari per colpire la nave? (Cinque al massimo).

Questo metodo è chiamato 'ricerca binaria' perché divide il problema sempre in due parti.

---

## Battaglia navale – *Un gioco a ricerca Hash*

---

### Istruzioni

1. Si prende un foglio come nei giochi precedenti dicendo al compagno il numero della nave scelta.
2. In questo gioco troverete a quale colonna (da 0 a 9) la nave appartiene. Per far ciò, occorre semplicemente sommare le cifre del numero della nave. L'ultima cifra della somma identifica la colonna. Per esempio, per localizzare la nave il cui numero è 2345, occorre sommare  $2+3+4+5$ , il cui risultato è 14. L'ultima cifra è 4, perciò la nave deve essere nella quarta colonna. Una volta identificata la colonna occorre trovare quale nave della colonna è quella giusta. Questa tecnica è chiamata 'hash' (che in inglese suona come mischiare) perché le cifre sono schiacciate assieme.
3. Giocate usando questa nuova strategia. Si può sfruttare il medesimo foglio per molte volte, basta scegliere la nave da colonne diverse.

(Al contrario degli altri giochi, i fogli di riserva 3A' e 3B' devono essere usati in coppia, perché la distribuzione delle navi in colonna deve corrispondere).

### Discussione che segue

1. Accumulate e discutete i punteggi come prima.
2. Quali navi sono molto semplici da trovare? (Quelle che sono da sole nella colonna). Quali navi è complesso trovare? (Quelle la cui colonna ne contiene molte).
3. Quale tra le strategie di ricerca viste è la più veloce? Perché?

Quali sono i vantaggi per ciascuna delle tre modalità di ricerca viste? (Il secondo modo è più veloce del primo, ma il primo non richiede che le navi siano in ordine. Il terzo modo è in genere più veloce degli altri due, ma qualche volta potrebbe essere veramente lento. Nel peggiore dei casi, se tutte le navi sono nella medesima colonna, il terzo metodo è pari al primo, in termini di velocità di ricerca).

---

# Attività aggiuntive

---

1. I bambini possono disegnarsi da soli i fogli per la battaglia usando uno dei tre formati. Per il secondo gioco, occorre che mettano le navi in ordine ascendente. Chiedete come possono far diventare il gioco Hash molto complesso. (Il più difficile è quando tutte le navi sono nella medesima colonna). E come lo si può far diventare il più semplice? (Occorre inserire lo stesso numero di navi in ciascuna colonna).
  2. Cosa succede se la nave che si stava cercando non era dove ipotizzato? (Nel gioco lineare impiegherete 26 colpi per trovarla. In quello binario 5. quando si usa quello Hash il numero di colpi dipende da quante navi ci sono nella colonna).
  3. Usando la strategia a ricerca binaria quanti colpi sono necessari per trovare una nave tra qualche centinaio (circa 6), qualche migliaio (circa nove), oppure un milione (circa diciannove)? (Attenzione che il numero di colpi aumenta molto lentamente in confronto al numero delle navi. Solo un colpo in più è richiesto quando il numero delle navi raddoppia, perciò il numero di colpi è proporzionale al logaritmo del numero delle navi totali).
-

***Le mie navi***

***Numero di colpi sparati:***

A	B	C	D	E	F	G	H	I	J	K	L	M
9058	7169	3214	5891	4917	2767	4715	674	8088	1790	8949	13	3014
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
8311	7621	3542	9266	450	8562	4191	4932	9462	8423	5063	6221	7244

***Le tue navi***

***Numero di colpi sparati:***

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

1A

**Le mie navi**

1630	9263	4127	405	4429	7113	3176	4015	7976	88	3465	1571	8625
A	B	C	D	E	F	G	H	I	J	K	L	M
2587	7187	5258	8020	1919	141	4414	3056	9118	717	7021	3076	3336
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

**Numero di colpi sparati:**

**Le tue navi**

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

**Numero di colpi sparati:**

**1B**

**Le mie navi**

163	445	622	1410	1704	2169	2680	2713	2734	3972	4208	4871	5031
A	B	C	D	E	F	G	H	I	J	K	L	M
5283	5704	6025	6801	7440	7542	7956	8094	8672	9137	9224	9508	9663
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

**Numero di colpi sparati:**

**Le tue navi**

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

**Numero di colpi sparati:**

**2A**

*Le mie navi*

*Numero di colpi sparati:*










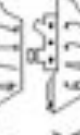








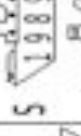
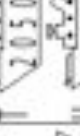




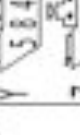

33	183	730	911	1927	1943	2200	2215	3451	3519	4055	5548	5655
A	B	C	D	E	F	G	H	I	J	K	L	M
5785	5897	5905	6118	6296	6625	6771	6831	7151	7806	8077	9024	9328
N	O	P	Q	R	S	T	U	V	W	X	Y	Z



























*Le tue navi*

*Numero di colpi sparati:*

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

**2B**

<b>Le mie navi</b>		<b>Numero di colpi sparati:</b>																	
0	A  B 	1	C  D 	2		3	E  F  G 	4	H  I  J  K 	5	L  M  N 	6	O  P  Q 	7	R  S  T  U 	8	V 	9	W  X  Y  Z 

<b>Le tue navi</b>		<b>Numero di colpi sparati:</b>																	
0	A  B  C  D 	1	E  F  G 	2	H  I  J 	3	K 	4	L  M  N 	5		6	O  P  Q 	7	R  S  T  U 	8	V  W  X 	9	Y  Z 

3 A



<b>Le mie navi</b>										<b>Numero di colpi sparati:</b>																																																																																									
0	A 9308	E 6519	H 1524	K 4135	L 9050	O 4200	R 3121	V 2385	Y 1990	1	B 1478	F 2469	I 8112	M 1265	P 7153	S 9503	T 1114	W 5832	Z 2502	2	C 8417	G 5105	J 2000	N 5711	Q 6028	U 7019	X 1917			3	D 9434									4										5										6										7										8										9									

<b>Le tue navi</b>										<b>Numero di colpi sparati:</b>																																																																																									
0	A	B	C	D						1										2										3										4										5										6										7										8										9									

# 3B

**Le mie navi**

**Numero di colpi sparati:**

6123	1519	9024	5164	2038	2142	7156	9974	9375	7104	1004	1023	5108
A	B	C	D	E	F	G	H	I	J	K	L	M
1884	3541	5251	4840	3289	3654	2480	5602	8965	4053	2405	2304	1959
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

**Le tue navi**

**Numero di colpi sparati:**

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

**1A'**

**Le mie navi**

**Numero di colpi sparati:**

2387	9003	3951	5695	1284	4761	7118	1196	1741	3791	3405	3132	6682
A	B	C	D	E	F	G	H	I	J	K	L	M
9493	9864	7359	1250	7036	2916	7562	9299	8910	6713	5173	8617	4222
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

**Le tue navi**

**Numero di colpi sparati:**

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

**1B'**

**Le mie navi**

**Numero di colpi sparati:**

28	326	943	1321	1896	2346	2430	2929	3106	3417	4128	4717	4915
A	B	C	D	E	F	G	H	I	J	K	L	M
5123	5615	6100	7015	7120	7695	7812	8103	8719	9020	9608	9713	9911
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

**Le tue navi**

**Numero di colpi sparati:**

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

**2A'**

**Le mie navi**

**Numero di colpi sparati:**

56	194	306	1024	1510	1807	2500	2812	3011	3902	4178	5902	5915
<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>	<b>I</b>	<b>J</b>	<b>K</b>	<b>L</b>	<b>M</b>
6102	6526	6818	7020	7155	7913	8016	8230	8599	8902	9090	9526	9812
<b>N</b>	<b>O</b>	<b>P</b>	<b>Q</b>	<b>R</b>	<b>S</b>	<b>T</b>	<b>U</b>	<b>V</b>	<b>W</b>	<b>X</b>	<b>Y</b>	<b>Z</b>

**Le tue navi**

**Numero di colpi sparati:**

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>	<b>I</b>	<b>J</b>	<b>K</b>	<b>L</b>	<b>M</b>
<b>N</b>	<b>O</b>	<b>P</b>	<b>Q</b>	<b>R</b>	<b>S</b>	<b>T</b>	<b>U</b>	<b>V</b>	<b>W</b>	<b>X</b>	<b>Y</b>	<b>Z</b>

**2B'**

<b>Le mie navi</b>										<b>Numero di colpi sparati:</b>											
0	A 1982	B 7841	1	C 6113	D 1055	2		3	E 9121 F 1011 G 7984	4	H 5009 I 2551 J 1751 K 4848	5	L 1248 M 1716 N 2148	6	O 2004 P 5173 Q 2805	7	R 9369 S 1321 T 3004 U 7190	8	V 3285	9	W 9172 X 2052 Y 6012 Z 7525

<b>Le tue navi</b>										<b>Numero di colpi sparati:</b>																													
0	A	B	C	D	1	E	F	G		2	H	I	J		3	K		4	L	M	N	5		6	O	P	Q	7	R	S	T	U	8	V	W	X	9	Y	Z

# 3A'

<b>Le mie navi</b>										<b>Numero di colpi sparati:</b>																																																																																									
0	A 8615	E 1361	H 7726	K 3000	L 1814	O 9656	R 6993	V 8208	Y 2917	1	B 7003	F 7644	I 9003	M 2002	N 8844	P 4002	S 3121	W 9423	Z 4122	2	C 1991	G 5600	J 5557			Q 1221	T 4300	X 4176		3	D 6211						U 1907			4										5										6										7										8										9									

<b>Le tue navi</b>										<b>Numero di colpi sparati:</b>																																																																																									
0	A	C	D							1										2										3	E	F	G							4	H	I	J	K						5	L	M	N							6	O	P	Q							7	R	S	T	U						8										9	W	X	Y	Z					

# 3B'

# Cosa c'entra tutto questo?

---

I computer memorizzano molte informazioni e hanno bisogno di rintracciarle spesso e velocemente. Uno dei problemi più grossi legati alle ricerche è relativo ai motori di ricerca di Internet i quali devono ricercare miliardi di pagine in frazioni di secondo. I dati che un computer deve ricercare, come ad esempio una parola, un bar code o il nome di un autore, sono chiamati chiavi di ricerca (*search key*).

I computer riescono ad elaborare l'informazione molto velocemente e si potrebbe pensare che partano dall'inizio della loro memoria continuando a cercare fintantoché l'informazione voluta non è trovata. Questo, in effetti, è quello che si fa con il gioco a *ricerca lineare*. Ma il metodo è molto lento —anche per i computer. Si supponga per esempio che un supermarket abbia 10,000 prodotti differenti sugli scaffali. Quando un codice a barre è scannerizzato, il computer potrebbe dare un'occhiata a tutti quei 10,000 numeri per trovare il nome e il prezzo del prodotto. Anche se ciò comportasse solo un millesimo di secondo per ciascun prodotto, sarebbe necessari 10 secondi per passare l'intera lista. Immaginatevi quanto ci si impiegherebbe per passare alla cassa tutta la spesa della famiglia!

Una strategia migliore è la *ricerca binaria*. Con questo metodo i numeri sono messi in ordine. Il controllo di quello che sta in mezzo alla lista identificherà la metà nella quale la chiave di ricerca è presente. Il procedimento è ripetuto fintantoché l'articolo è trovato. Ritornando all'esempio del supermarket i 10,000 articoli possono essere ricercati solo con quattordici tentativi e con un tempo che si aggira su un centesimo di secondo, difficilmente percepibile.

Una terza strategia per cercare le informazioni è denominata *hash*. In questo caso la chiave di ricerca è manipolata per indicare esattamente dove trovare l'informazione cercata. Per esempio, se la chiave di ricerca è un numero telefonico, potreste sommare tutte le cifre del numero e calcolare il resto diviso 11. ciò facendo si arriva grosso modo a quanto discusso nell'Attività 4 – un piccolo frammento di dato il cui valore dipende da tutti gli altri. Di norma, il computer troverà velocemente quello che sta cercando senza alcun problema. Rimane comunque una piccola probabilità che molte chiavi di ricerca vadano a finire nella medesima locazione, nel qual caso il computer dovrà andare a fare una ricerca in quella lista finché non trova l'informazione desiderata.

I programmatori di solito usano un qualche adattamento della strategia hash per la ricerca, a meno che non sia importante mantenere i dati in ordine o a meno che una risposta lenta di tanto in tanto sia non accettabile.



# Attività 7

---

## Il piccolo e il grande—*Algoritmi di ordinamento*

### Sommario

I computer sono spesso usati per ordinare un elenco di oggetti, come per esempio nomi in ordine alfabetico, appuntamenti o e-mail per data, numeri in ordine numerico. Si è visto come l'ordinamento ci permette di trovare facilmente l'oggetto che ricerchiamo facendo anche in modo di vedere a colpo d'occhio i valori estremi di qualsiasi lista. Per esempio, se si ordinano i risultati di un test allora il risultato più basso e quello migliore diventano evidenti.

Se si usa un metodo improprio, ci si può impiegare molto tempo per ordinare una lunga lista di oggetti, anche se si usa un computer veloce ed efficiente. Per fortuna, però, esistono molti metodi conosciuti per effettuare tale operazione. In questa attività, i bambini scopriranno diversi modi per effettuare l'ordinamento e verificheranno come un buon metodo possa completare il compito molto più efficientemente e speditamente di un metodo semplice.

### Abilità

- ✓ Usare bilance
- ✓ Ordinare
- ✓ Confrontare

### Età

- ✓ A partire da 8 anni

### Materiali

Ciascun gruppo di bambini avrà:

- ✓ insiemi di 8 contenitori della stessa dimensione ma con pesi diversi (ad esempio, cartoni del latte o scatole dei rotoli di pellicola o d'alluminio riempiti di sabbia)
- ✓ Bilance
- ✓ Foglio di lavoro: ordinare i pesi (pagina 75)
- ✓ Foglio di lavoro: dividi et impera (pagina 76)

# Il piccolo e il grande

---

## **Discussione**

I computer hanno spesso la necessità di ordinare liste di oggetti. Allo stesso tempo, per noi, pensare a tutti i luoghi ove riporre e ritrovare gli oggetti ordinati è altrettanto importante. Cosa accadrebbe se questi oggetti non fossero in ordine?

I computer solitamente confrontano solo due oggetti alla volta. L'attività alla pagina successiva usa questa restrizione per dare ai bambini un'idea di cosa significhi ordinare oggetti.

## **Attività**

1. Dividete in gruppi di bambini.
2. Ciascun gruppo avrà bisogno di una copia della pagina dell'attività a pagina 75 oltre a pesi e bilance.
3. Lasciate che i bambini facciano l'attività, poi discutete assieme.

## Attività del foglio di lavoro: ordinare pesi

**Scopo:** trovare il miglior metodo per ordinare un insieme di pesi.

**Avrai bisogno di:** sabbia o acqua, 8 contenitori identici, bilance

**Cosa fare:**

1. Riempite ciascun contenitore con un diverso quantitativo di sabbia o acqua. Chiudete bene.
2. Mescolate i contenitori così che non si possa riconoscere il loro peso.
3. Trovate il contenitore più leggero. Qual è il sistema più facile di farlo?

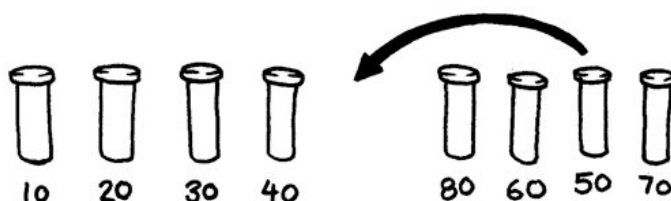
**Nota:** Potete usare solo la bilancia per pesare ciascun contenitore. Solo due contenitori possono essere confrontati alla volta.

4. Scegliete a caso 3 contenitori e metteteli in ordine dal più leggero al più pesante usando solamente la bilancia. Come potete fare? Qual è il minimo numero di confronti necessario? Perché?
5. Adesso mettete tutti gli oggetti in ordine dal più leggero al più pesante.

Quando ritenete di aver concluso, controllate il risultato confrontando tra loro con la bilancia due oggetti vicini.

### Selection Sort (*ordinamento per selezione*)

Un metodo che si può usare con il computer è chiamato *selection sort*. Funziona così. Trovate prima l'oggetto più leggero e riponetelo da un lato. Successivamente, trovate il più leggero tra gli oggetti rimasti e mettetelo vicino a quello più leggero trovato precedentemente. Ripetete questa operazione fino a che non avete terminato gli oggetti da pesare.



Contate quanti confronti avete effettuato.

Extra per Esperti: mostrate come potete calcolare matematicamente quanti passi sono necessari per ordinare 8 oggetti? E 9 oggetti? 20?

## Attività del foglio di lavoro: dividi et impera

### Quicksort

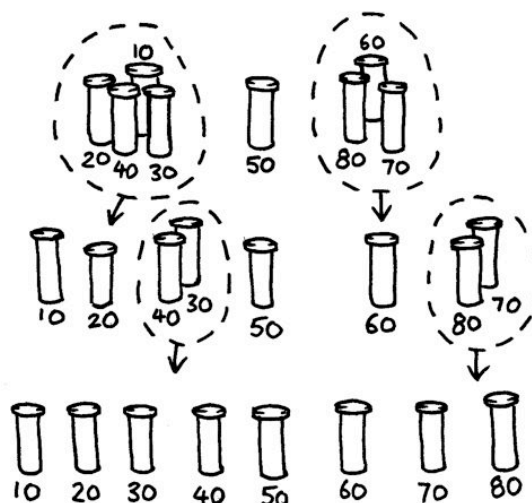
Quicksort è molto più veloce di selection sort, in particolare quando si hanno insiemi di oggetti molto grandi. Infatti, è uno dei migliori metodi di ordinamento conosciuti. Funziona così.

Scegliete a caso un oggetto e posizionate su un piatto della bilancia.

Ora confrontate ciascun oggetto rimanente con quello precedentemente selezionato. Mettete quelli più leggeri a sinistra, l'oggetto selezionato per primo al centro, e quelli più pesanti a destra. (Si potrebbero anche avere molti più oggetti da un alto rispetto all'altro).

Scegliete uno degli insiemi di oggetti a destra o sinistra e ripetete le precedenti operazioni. Fate lo stesso per l'altro insieme. Ricordatevi di mantenere al centro l'oggetto che selezionate per primo.

Continuate a ripetere questa procedura in tutti i gruppi fino a che ciascun gruppo contiene solo un oggetto. A questo punto, gli oggetti saranno in ordine dal più leggero al più pesante.



Quanti confronti sono necessari in questo procedimento?

Quicksort è più efficiente di selection sort a meno che non accada di iniziare con l'oggetto più leggero o pesante in assoluto. Se siete fortunati da aver scelto l'oggetto con il peso intermedio, dovrete aver effettuato solo 14 confronti, la metà di quelli effettuati con il selection sort. In ogni caso, quicksort non sarà peggiore di selection sort e, anzi, può essere molto molto più efficiente!

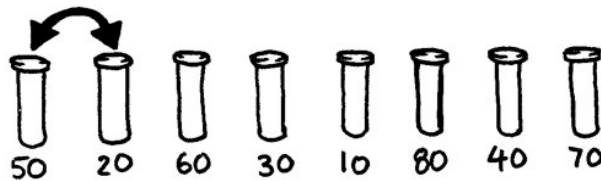
Extra per Esperti: Se quicksort incidentalmente scegliesse sempre l'oggetto più leggero, quanti confronti effettuerebbe?

## Variazioni ed estensioni

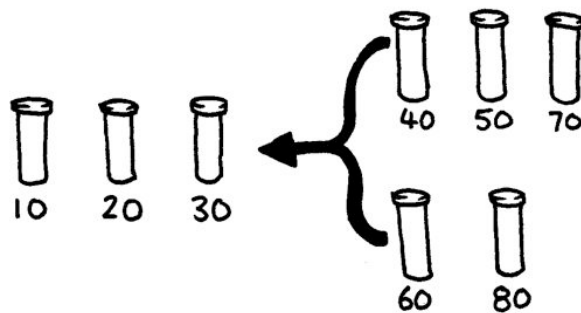
Sono stati inventati molti sistemi diversi per l'ordinamento. Potete usare anche i seguenti:

*Insertion sort* (ordinamento per inserimento) lavora rimuovendo ciascun oggetto da un insieme non ordinato e inserendolo nella posizione corretta in un altro insieme (come da figura). Ad ogni inserzione, l'insieme di oggetti non ordinati si riduce e quello ordinato cresce, fino a che l'intero insieme di oggetti originario è ordinato. I giocatori di carte usano molto spesso questo sistema per ordinare *una mano* di carte.

*Bubble sort* (ordinamento a bolle) opera nell'insieme di oggetti molte volte, scambiando ogni oggetto di posto, da destra a sinistra e viceversa quando incontra un oggetto a destra più leggero di quello di sinistra. L'insieme di oggetti è ordinato quando non sono più necessari scambi. Questo metodo non è molto efficiente, ma molti trovano che sia molto più facile da capire di altri.



*Mergesort* (ordinamento per fusione) è un altro metodo che usa 'dividi et impera' per ordinare una lista di oggetti. Per prima cosa la lista è divisa casualmente in due liste con un numero di oggetti uguale (se gli oggetti sono in numero dispari, una delle due nuove liste conterrà un oggetto in più). Ciascuna delle due sotto-liste viene ordinata e alla fine entrambe vengono *fuse* assieme. Fondere le due sotto-liste è un procedimento semplice: basta rimuovere ripetutamente il più piccolo oggetto scelto tra i due all'inizio delle sotto-liste ordinate. Nella figura sottostante, gli oggetti con peso 40 e 60 grammi sono all'inizio delle due liste, così l'oggetto da prelevare e da *fondere* nella lista finale che si sta costruendo (a sinistra) è l'oggetto da 40 grammi. Come si ordinano le sotto-liste? Semplice, basta usare... Mergesort! Così facendo, alla fine tutte le sotto-liste saranno composte solo da un unico elemento, pertanto non ci si deve preoccupare di sapere quando finire....



## Cosa c'entra tutto questo?

L'informazione è molto più facile da trovare in un insieme ordinato. Gli elenchi telefonici gli indici dei libri usano l'ordine alfabetico e la vita sarebbe molto più complessa se non

lo facessero...! se una lista di numeri è ordinata, i casi agli estremi sono più facili da notare perché questi sono all'inizio o alla fine della lista. I casi duplicati sono anch'essi facili da notare, perché sono vicini tra loro.

I computer passano molto del loro tempo a mettere in ordine oggetti, così gli informatici hanno trovato metodi veloci ed efficienti per farlo. Alcuni dei metodi più lenti, come insertion sort, selection sort o bubble sort possono essere utili in situazioni speciali, ma sono generalmente usati quelli più veloci come quicksort.

Quicksort sfrutta un concetto chiamato *ricorsione*. Questo significa che si continua a dividere una lista di oggetti in parti più piccole applicando lo stesso tipo di ordinamento alla parti piccole. Questo approccio in particolare è denominato *dividi et impera*. La lista è ripetutamente suddivisa finché è sufficientemente piccola da *conquistare (impera)*. Per quicksort, le liste sono suddivise finché non contengono un solo elemento. È banale ordinare un solo elemento! Sebbene sembri molto laborioso, questo metodo è incredibilmente più veloce di altri.

# Soluzioni e suggerimenti

---

1. il modo migliore di trovare l'oggetto più leggero è quello di considerare tutti gli oggetti uno alla volta tenendo traccia del più leggero fino a quel momento. Cioè, confrontate due oggetti e tenere il più leggero, ripetendo il metodo fino a che tutti gli oggetti siano stati considerati.
2. Confrontate i pesi su una bilancia. Ciò può essere fatto facilmente con 3 pesate, e qualche volta 2—se i bambini comprenderanno che il confronto è un operatore transitivo (cioè, se A è più leggero di B e B è più leggero di C, allora A dev'essere più leggero di C).

## Esperti:

Ecco un suggerimento per calcolare il numero totale di passi di confronto che sono effettuati da selection sort .

Per trovare il minimo di due oggetti effettuerete un solo confronto, per 3 oggetti ne effettuerete 2, 3 confronti per 4 oggetti e così via. Per ordinare 8 oggetti usando selection sort avrete bisogno di 7 confronti per trovare il primo oggetto (il più leggero), 6 per il secondo, 5 per il terzo e così via:

$$7 + 6 + 5 + 4 + 3 + 2 + 1 = 28 \text{ confronti}$$

$n$  oggetti necessiteranno di  $1 + 2 + 3 + 4 + \dots + n - 1$  confronti per essere ordinati.

Sommare questi numeri diviene più facile se si raggruppano diversamente.

Per esempio, per sommare i numeri  $1 + 2 + 3 + \dots + 20$ , raggruppateli così:

$$\begin{aligned} &(1 + 20) + (2 + 19) + (3 + 18) + (4 + 17) + (5 + 16) + \\ &(6 + 15) + (7 + 14) + (8 + 13) + (9 + 12) + (10 + 11) \\ &= 21 \times 10 \\ &= 210 \end{aligned}$$

3. In generale, sarà  $1 + 2 + 3 + 4 \dots + n - 1 = n(n - 1)/2$ .

# Attività 8

---

## ***Batti il tempo—Reti di ordinamento***

### **Sommario**

Anche se i computer sono veloci, c'è un limite a quanto rapidamente riescono a risolvere i problemi. Un modo per accelerare ulteriormente il tempo di soluzione è quello di usare molti computer alla volta ciascuno dei quali risolve una parte del problema. In questa attività useremo le *reti di ordinamento* per capire come effettuare più di un confronto nello stesso istante.

### **Abilità**

- ✓ Confrontare
- ✓ Ordinare
- ✓ Sviluppare algoritmi
- ✓ Risolvere problemi assieme

### **Età**

- ✓ A partire da 7 anni

### **Materiali**

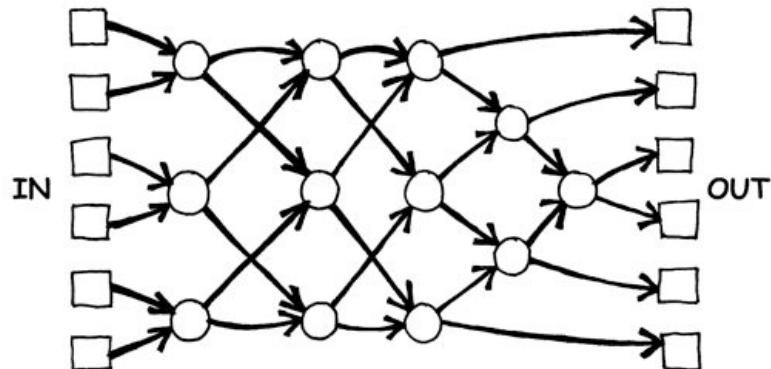
Questa attività si può effettuare all'aperto.

- ✓ Gesso
- ✓ Due insiemi di 6 carte.  
Ricopiate il foglio principale: reti di ordinamento (pagina 82) in un foglio e ritagliatelo
- ✓ Un cronometro



# Reti di ordinamento

Prima di iniziare l'attività usate il gesso per disegnare su una superficie esterna lo schema seguente:

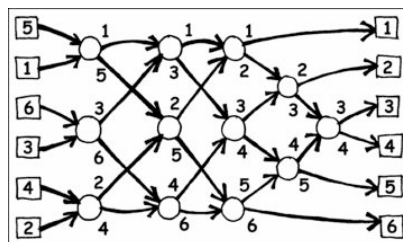


## Istruzioni per i bambini

Questa attività vi mostrerà come i computer ordinano numeri casuali sfruttando la cosiddetta rete di ordinamento.

1. Organizzatevi in gruppi di 6. Solo un gruppo alla volta potrà giocare.
2. Ogni giocatore deve avere una carta numerata.
3. Ogni giocatore deve sostare nel quadrato a sinistra (IN) dello schema disegnato. I giocatori con i numeri devono essere mescolati.
4. Muovetevi lungo le linee e quando raggiungete un cerchio **dovete attendere che arrivi un compagno.**
5. Quando il compagno arriva, confrontate le vostre carte. La persona con il numero più piccolo prosegue verso sinistra, l'altro verso destra.
6. Siete nell'ordine giusto quando arrivate all'altro capo dello schema?

Se una squadra compie un errore, i bambini devono re-iniziare il gioco. Controllare se avete ben compreso l'operazione da effettuare nei cerchi dello schema, dove il valore più piccolo prosegue a sinistra e il più grande a destra. Per esempio:



## Foglio da fotocopiare: reti di ordinamento

---

**1**

**2**

**3**

**4**

**5**

**6**

---

**156**

**221**

**289**

**314**

**422**

**499**

## Variazioni

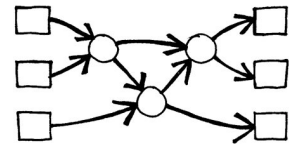
1. Quando i bambini hanno raggiunto familiarità con l'attività, usate il cronometro per misurare quanto tempo ciascuna squadra impiega per concludere il gioco.
2. Usate carte con numeri più grandi (ad esempio, quelli presenti nella fotocopia principale).
3. Ideate carte con numeri ancora più grandi che siano complesse da confrontare, oppure usate parole e confrontatele alfabeticamente.

## Attività aggiuntive

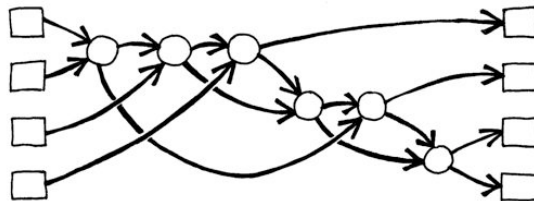
1. Cosa succede se il numero più piccolo va a destra invece di sinistra e viceversa? (I numeri saranno ordinati in ordinamento inverso)

Lo schema funziona lo stesso se si va al contrario? (Non funziona, e i bambini dovrebbero essere in grado di trovare un esempio di input che non viene ordinato correttamente)

2. Cercate di progettare una rete di ordinamento più piccola o più grande. Per esempio, ecco una rete che ordina solo 3 numeri. I bambini dovrebbero riuscire ad idearla da soli.



3. Seguono due diverse reti di ordinamento che effettuano l'operazione su 4 numeri. Quale delle due è più veloce? (La seconda. Mentre la prima richiede che tutti i confronti siano effettuati in sequenza, uno dopo l'altro, la seconda prevede che qualche operazione sia effettuata contemporaneamente. La prima rete è un esempio di calcolo seriale, la seconda sfrutta il calcolo parallelo per andare più veloce).



4. Cercate di progettare una rete ancora più grande..
5. Le reti possono anche essere usate per trovare il valore minimo dato in input. Per esempio, a destra è disegnata una rete per 8 valori: l'unico valore di uscita sarà quello minimo (gli altri valori rimangono intrappolati nella rete).

Quali processi di tutti i giorni potrebbero o non potrebbero essere accelerati usando il parallelismo? Per esempio, cucinare un pasto sarebbe più lento se usassimo solo un fornello perché tutti i piatti dovrebbero essere cotti uno dopo l'altro. Quali lavori potrebbero essere completati più velocemente impiegando più persone? E quali no?

# Cosa c'entra tutto questo?

---

Via via che si usano sempre più i computer si vorrebbe che riuscissero ad elaborare sempre più velocemente le informazioni.

Un modo per incrementare la velocità di elaborazione è quello di scrivere programmi che usano meno passi per risolvere il problema (come mostrato nelle attività 6 e 7).

Un altro modo per risolvere i problemi più velocemente è quello di usare più computer nello stesso tempo, operanti su diverse parti dello stesso problema. Per esempio, nella rete di ordinamento con 6 numeri, sebbene siano necessari 12 confronti per ordinare i numeri, si possono effettuare fino a 3 comparazioni simultaneamente. Ciò significa che il tempo richiesto è solo di 5 passi di elaborazione. Questa rete parallela ordina la lista di numeri ad una velocità più che doppia rispetto ad un sistema che effettua un confronto per volta.

Non tutti i problemi possono essere risolti più velocemente usando il calcolo parallelo. Per analogia, immaginate una persona che faccia un fossato lungo 10 metri. Se 10 persone facessero un metro di quel fossato, l'intero lavoro sarebbe terminato più velocemente. Ma lo stesso non accadrebbe con uno scavo di 10 metri di profondità — il secondo metro non è accessibile finché non è terminato il primo. Gli informatici stanno ancora studiando qual è il miglior modo di suddividere i problemi così che possano essere elaborati da computer in parallelo.

---

# Attività 9

---

## *La città fangosa—Minimal Spanning Trees*

### **Sommario**

la nostra società ha molti collegamenti in rete: la rete telefonica, la rete energetica, la rete stradale. Per una rete in particolare, ci sono solitamente più scelte su come posizionare le strade, i cavi o i collegamenti radio. Dobbiamo quindi trovare i modi più efficienti per collegare gli oggetti tra loro.

### **Età**

- ✓ A partire da 9 anni

### **Abilità**

- ✓ Risolvere problemi

### **Materiali:**

Ogni bambino avrà bisogno di:

- ✓ Attività pratica: il problema della città fangosa (pagina 88)
  - ✓ Tavole o quadrati di cartone (circa 40 per bambino)
-

# La città fangosa

---

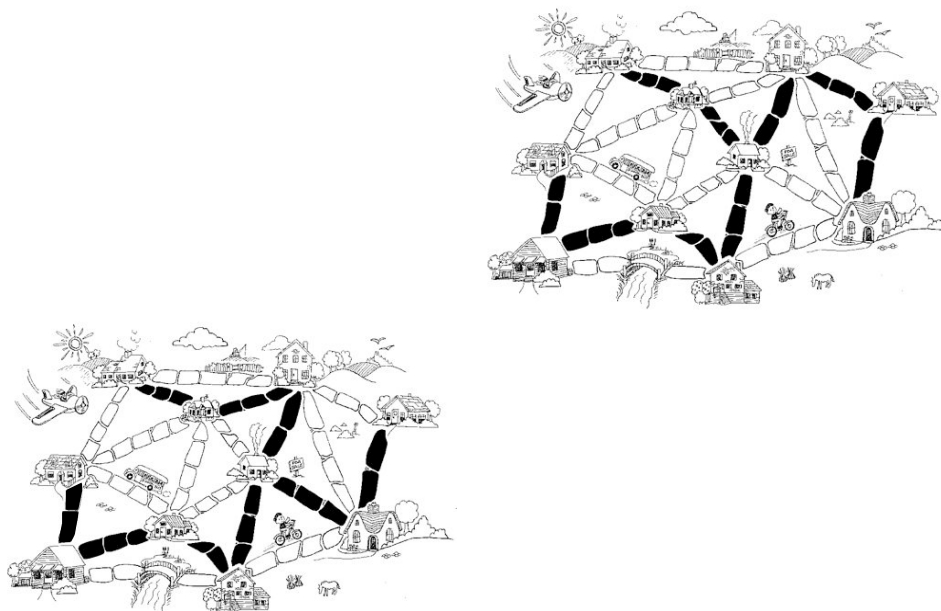
## Introduzione

Questa attività vi mostrerà come sono usati i computer per trovare la miglior soluzione in problemi quotidiani, come ad esempio quello di collegare linee elettriche tra case. I bambini devono usare il foglio di lavoro di pagina 88 che spiega il problema della ‘Città Fangosa’.

## Discussione che segue

Condividete la soluzione trovata dai bambini. Quale strategia hanno usato?

Una buona strategia di partenza è quella di partire con una mappa vuota e gradualmente posizionare le tavole finché tutte le case sono collegate tra loro, a partire dai collegamenti di lunghezza inferiore, facendo attenzione di non collegare case già tra loro collegate. Si possono avere diverse soluzioni se si scelgono diversi cammini di collegamento della stessa lunghezza. Due soluzioni alternative sono illustrate di seguito.



Un'altra strategia è quella di cominciare con tutti i cammini già selezionati rimuovendoli man mano se non sono necessari. In ogni caso, ciò richiede maggior sforzo della strategia precedente.

Dove potete trovate reti di questo tipo nella quotidianità?

Gli informatici chiamano la rappresentazione di queste reti con il nome di “grafi”. Le reti reali possono essere rappresentati da grafi per arrivare a risolvere problemi come ad esempio progettare il posizionamento migliore per una rete di strade tra città, oppure trovare il miglior modo di connettere delle città con voli aerei.

Ci sono anche molti algoritmi che possono essere applicati ai grafi, come ad esempio quello di trovare la distanza inferiore tra due punti, o il cammino più corto che tocca tutti i punti del grafo

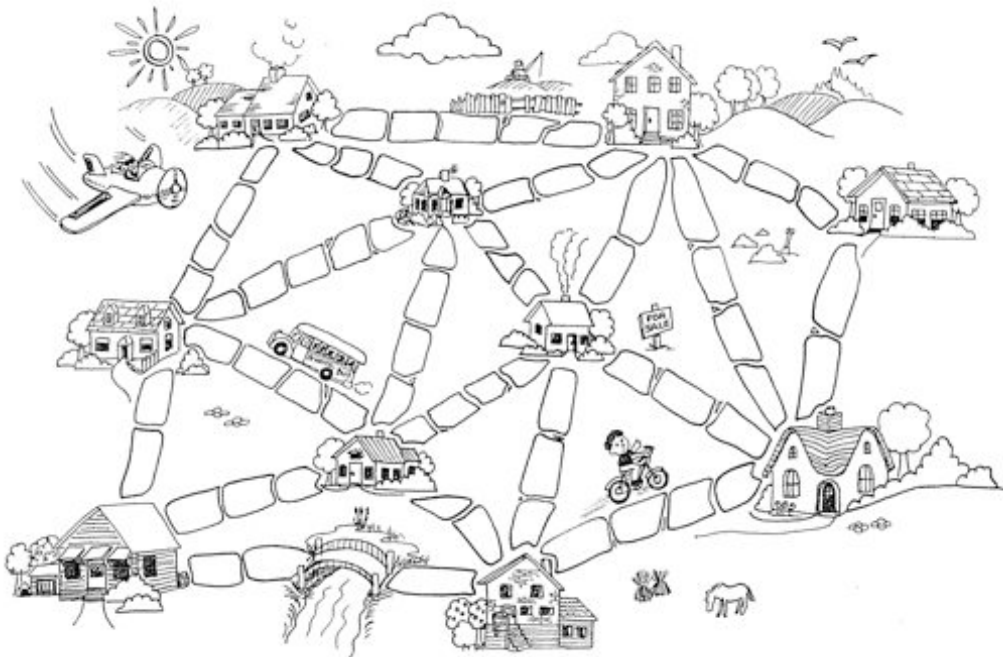
## Foglio di lavoro: Il Problema della Città Fangosa

Tanto tempo fa c'era una città che non aveva strade. Gironzolare nella città era particolarmente difficile dopo un temporale perché tutt'intorno c'era solo fango – le macchine si bloccavano nel fango e le persone si sporcavano sempre i loro stivali. Il sindaco della città decise così che qualche strada doveva essere pavimentata ma non voleva spendere più soldi del necessario perché voleva costruire anche una piscina. Il sindaco specificò allora due condizioni

1. devono essere pavimentate abbastanza strade perché sia possibile per ognuno andare dalla propria casa ad un'altra qualsiasi casa e,
2. la pavimentazione deve costare il meno possibile.

Quella che segue è la mappa della città. Il numero di tavole (pietre) tra le case rappresenta il costo di per pavimentare quel tratto. Trovate il miglior percorso che connette tutte le case, ma usate il minor numero di tavole (pietre) possibile. (il ponte non conta, non ha necessità di essere pavimentato).

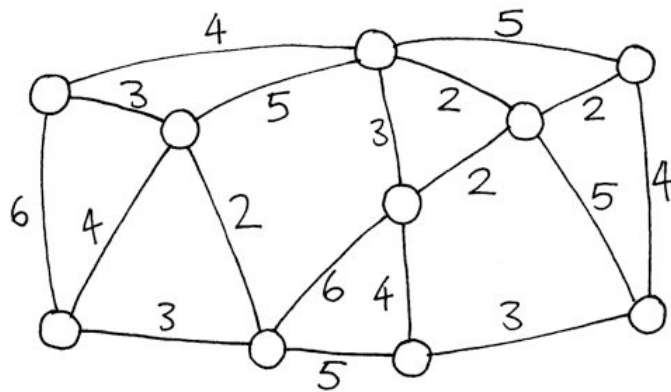
Quale strategia usereste per risolvere il problema?





### Variazioni ed estensioni

Quello che segue è un modo alternativo di rappresentare città e strade:



Le case sono rappresentate da cerchi, le strade fangose da linee, e la lunghezza delle strade da un numero vicino alla linea. .

Gli informatici e i matematici usano spesso questo diagramma per rappresentare problemi. Lo chiamano *grafo*. Questo potrebbe portare un po' di confusione poiché la parola assomiglia a quella usata in ambito statistico per rappresentare dati numerici cioè il “grafico”, ma sono due cose diverse, anche se il nome si assomiglia. In un grafo, le lunghezze delle linee può non essere correlata al numero che hanno vicino.

Costruite un vostro esempio di città fangosa e risolvetelo con i vostri compagni.

Potete trovare una regola che descriva quante strade o connessioni sono necessarie per la miglior soluzione? Dipende da quante case ci sono nella città?

# Cosa c'entra tutto questo?

---

Supponete di dover progettare come portare ad una nuova comunità l'energia, oppure il gas, o l'acqua. Una rete di fili o di tubi deve connettere tutte le case all'azienda elettrica o di fornitura dell'acqua. Ciascuna casa deve essere collegata alla rete in un qualche punto, e il percorso di collegamento tra la casa e l'azienda non importa più di tanto, importa che la casa sia collegata.

Il problema di progettare una rete con un percorso totale minimo è chiamato *minimal spanning tree*.

I *minimal spanning tree* non sono utili solo per il gas o l'elettricità; ci aiutano anche a risolvere problemi nelle reti di computer, nelle reti telefoniche, nelle condotte petrolifere, nelle rotte aeree. In ogni caso, quando, ad esempio, si decide quale rotta di viaggio è la migliore per una persona, occorre considerare non solo quanto costerà al viaggiatore, ma anche quanto costerà in generale. Nessuno vorrebbe passare ore in aereo prendendo la rotta più lunga perché è più economica. Il problema della città fangosa potrebbe non essere di molto aiuto per queste reti, perché calcola il percorso minimo *totale* delle strade o dei voli aerei.

I *Minimal spanning tree* sono utili anche come passo per risolvere altri problemi come il “problema del commesso viaggiatore” che cerca di trovare il percorso più corto per visitare ogni punto di una rete.

Ci sono algoritmi efficienti per risolvere i problemi *minimal spanning tree*. Un metodo semplice che conduce ad una soluzione ottima è quello di partire senza alcun collegamento, aggiungendoli a partire dai più corti e connettendo solo parti della rete che non sono ancora state connesse. Questo è chiamato algoritmo di Kruskal ( J.B. Kruskal, lo pubblicò nel 1955).

Per molti problemi sui grafi, incluso il “problema del commesso viaggiatore”, gli informatici devono ancora trovare metodi sufficientemente veloci per la miglior soluzione possibile.

## Soluzioni e suggerimenti

---

### **Variazioni ed estensioni** (pagina 89)

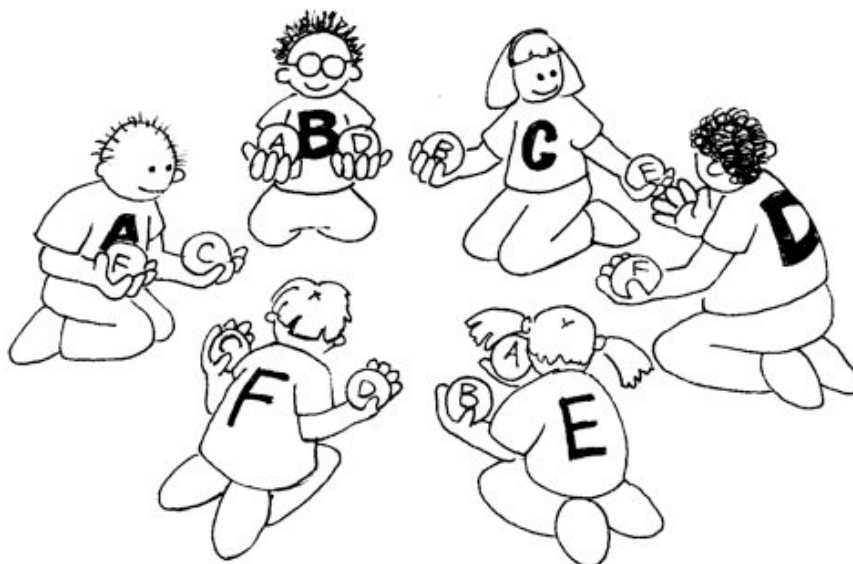
Quante strade o collegamenti sono necessari se ci sono  $n$  case in città? Ne deriva che una soluzione ottima avrà esattamente sempre  $n-1$  collegamenti, perché ciò è sufficiente per collegare  $n$  case e un collegamento in più creerebbe rotte alternative tra di esse.

---

# Attività 10

---

## Il gioco dell'arancia—*Instradamento e deadlock nelle reti.*



### Sommario

Quando ci sono molte persone che devono usare insieme una risorsa (per esempio le automobili che usano una strada o i messaggi in Internet) c'è la possibilità che le persone si blocchino a vicenda. Si dice in questo caso che il sistema è in deadlock (qualche volta tradotto in italiano come stallo). Occorre che le persone (o i computer) collaborino attivamente perché questo non succeda.

### Competenze richieste

- ✓ Idee di soluzione cooperativa di problemi
- ✓ Capacità di ragionamento logico

### Età

- ✓ a partire da 9 anni

### Materiale

Ogni studente deve avere:

- ✓ Due arance o palle da tennis
  - ✓ Etichette
-

# Il gioco dell'arancia

---

## Introduzione

Questo è un esempio di soluzione cooperativa di un problema. Si lavora a gruppi, ogni studente sarà contrassegnato da una lettera. Lo scopo è che ogni studente alla fine abbia le arance che recano l'etichetta con la propria lettera.

1. Dividere gli studenti in gruppi di cinque o più. Ogni gruppo si deve disporre in cerchio.
2. A ogni studente deve essere attribuita una lettera (unica all'interno del gruppo) che lo studente scriverà su una etichetta da tenere ben visibile (per esempio attaccandola alla maglietta). Se ci sono si possono usare i porta cartellini di riconoscimento. Preparate per ogni studente due arance con l'indicazione della lettera relativa. Uno studente solo per gruppo deve avere una arancia con la propria lettera invece che due (le arance verranno distribuite agli studenti una per mano, nel gruppo dovrà rimanere esattamente una mano libera).
3. Distribuite ora casualmente le arance agli studenti del gruppo. Ogni studente avrà quindi due arance tranne uno che avrà una mano libera. Fate in modo però che nessuno studente abbia arance con la sua lettera.
4. Gli studenti devono passarsi le arance fino a quando non si giunga alla soluzione, cioè che ogni studente abbia le arance corrispondenti alla propria lettera. Occorre seguire queste regole.
  - a) Non si può tenere più di un'arancia in ogni mano.
  - b) Un'arancia può essere passata nella mano libera dello studente immediatamente alla propria sinistra o alla propria destra nel cerchio. (Lo studente può passare una qualsiasi delle proprie arance al vicino).

Gli studenti impareranno presto che se tenteranno di usare politiche greedy (cioè tengono le arance non appena hanno quella corrispondente alla propria lettera) allora il gruppo potrebbe non riuscire ad arrivare alla soluzione del problema. Greedy letteralmente significa "avido, ingordo". Nell'inglese informatico si chiamano metodi di soluzione greedy quelli che tentano di trovare la soluzione in modo incrementale passando per soluzioni parziali che, una volta trovate, non vengono più messe in discussione. Non sempre i metodi greedy raggiungono la soluzione o non trovano la soluzione migliore.

È necessario forse sottolineare che non sono i singoli studenti che vincono il gioco quando hanno le arance con la propria lettera ma il gruppo ha risolto il problema quando tutti hanno le arance corrette.

## Discussione sul problema

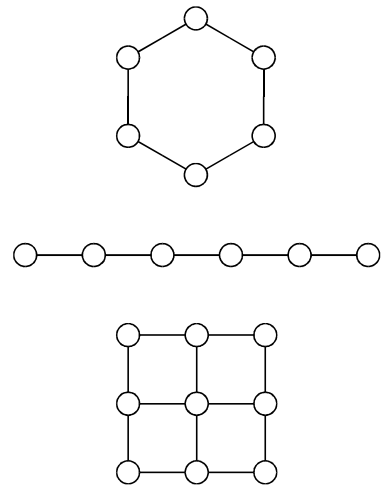
Quali strategie sono state usate dagli studenti per risolvere il problema?

Quando nella vita reale siete rimasti vittime di un deadlock? (Alcuni esempi possono essere ingorghi stradali, uso delle porte girevoli con molte persone che vogliono entrare e molte che vogliono uscire,....)

### Possibili estensioni

Provate l'attività con gruppi più o meno ampi.

- Chiedete agli studenti di inventare e provare regole diverse.
- Fare in modo che gli studenti tentino di risolvere il problema senza comunicare, magari seguendo una regola che hanno deciso a priori.
- Provare diverse configurazioni come essere disposti in fila, o in altre disposizioni dove alcuni studenti abbiano più di due vicini, come si può vedere nella figura qui a lato.



# Cosa c'entra tutto questo?

---

Instradamento e il deadlock sono problemi in molte reti, come le reti viarie, telefoniche, e reti informatiche come Internet. I tecnici spendono molto tempo per capire come risolvere questi problemi e come progettare le reti in modo che questi problemi siano facili da risolvere.

L'instradamento errato, le congestioni e il deadlock possono causare problemi frustranti in molte differenti reti. Pensate solamente al traffico nelle ore di punta. A New York è successo molte volte che il traffico fosse così congestionato che finisse in un vero deadlock: nessuno era più in grado di muovere la propria auto! Talvolta i computer sono "giù" nelle aziende, nelle banche o negli uffici pubblici a causa di un deadlock nelle comunicazioni. Progettare le reti in modo che l'instradamento delle comunicazioni sia efficiente e veloce e in modo da minimizzare le congestioni è un problema difficile per ingegneri di vari settori.

Talvolta più di una persona vuole lo stesso dato allo stesso tempo. Se uno specifico dato (come il saldo di un conto bancario) deve essere aggiornato, è importante bloccarlo (operazione di lock) durante l'aggiornamento. Se non viene bloccato, un altro utente potrebbe tentare di aggiornarlo nello stesso tempo e il risultato finale potrebbe essere scorretto. Se invece il blocco richiesto da un programma interferisce con il blocco richiesto da un altro programma ci può essere deadlock. Nel caso del conto bancario pensiamo a due operazioni contemporanee, la prima trasferisce il denaro dal conto A al conto B e la seconda dal conto B al conto A, quindi entrambi devono bloccare sia il conto A sia il conto B. Se il primo programma blocca il conto A mentre il secondo blocca il conto B il sistema è in deadlock perché il primo programma aspetterà che il secondo rilasci il blocco sul conto B mentre il secondo per farlo sta aspettando che il primo lasci libero il conto A.

Uno degli sviluppi più interessanti nella evoluzione della struttura dei computer è l'avvento dei computer paralleli, dove centinaia o migliaia di processori lavorano insieme collegati da una rete per formare un solo potentissimo computer. Molti problemi come quello del gioco dell'arancia devono essere giocati continuamente su queste reti (ma molto più velocemente!) per poter far funzionare questi computer.

---

# Parte III

Dire ai computer cosa devono fare  
–*Rappresentare le procedure*

# Dire ai computer cosa devono fare

---

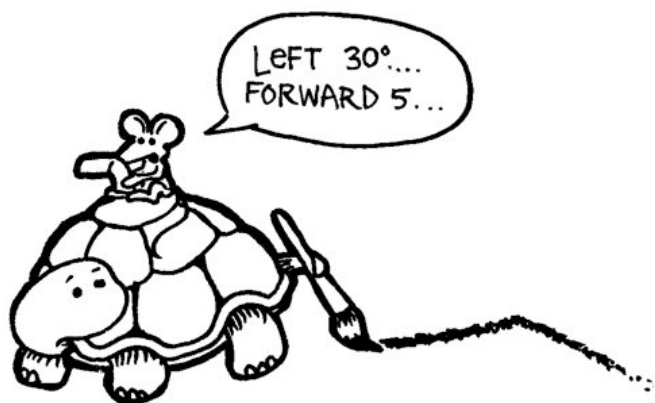
I computer seguono le istruzioni, milioni di istruzioni per ogni secondo. Per dire ad un computer cosa deve fare tutto ciò che dovete fare è dare le istruzioni giuste. Ma non è così semplice come sembra!

Quando ci vengono date istruzioni da seguire noi usiamo il buon senso e l'esperienza per interpretare quale possa essere il significato. Se qualcuno dicesse “passa attraverso quella porta”, chiaramente non intende che dobbiamo abbattere la porta ma che dobbiamo oltrepassare la porta ed è quindi necessario prima aprire la porta se questa è chiusa! I computer funzionano diversamente. Per esempio quando un computer controlla un robot occorre prendere le dovute precauzioni perché non possa causare danni e pericolo interpretando letteralmente le istruzioni, come l'ordine di passare attraverso una porta. Occorre un po' di tempo e di esperienza perché ci si possa abituare di avere a che fare con qualcosa che segue sempre le istruzioni alla lettera, senza “pensare”.

Le due attività di questa parte del testo servono per fornire alcune idee su come si possa comunicare con una macchina che obbedisce alla lettera usando un insieme prefissato di istruzioni.

La prima attività mostra il funzionamento di una “macchina” che i computer usano per riconoscere parole, numeri e stringhe (sequenze) di caratteri. Queste “macchine” sono chiamate “automi a stati finiti”.

La seconda attività ci introduce a come possiamo comunicare coi computer. Un buon programmatore deve imparare come parlare ai computer usando un prefissato insieme di istruzioni che verranno interpretate letteralmente. La sequenza delle istruzioni è il programma. Ci sono molti diversi linguaggi di programmazione fra i quali un programmatore può scegliere per scrivere le istruzioni, ma noi useremo un semplice linguaggio che si può scrivere senza un computer.



(a sinistra 30 gradi, avanti di 5....)

---



# Attività 11

---

## Caccia al tesoro—*Automi a stati finiti*

### Sommario

I programmi per computer spesso hanno necessità di elaborare sequenze di simboli come lettere o parole in un documento, o anche il testo di un altro programma. Gli scienziati informatici spesso usano un automa a stati finiti per questi compiti. Un automa a stati finiti (Finite State Automata o FSA in inglese) segue un insieme di istruzioni per vedere se il computer deve riconoscere la parola o la sequenza di simboli. Noi lavoreremo con qualcosa di equivalente a un automa a stati finiti: la mappa di un tesoro!

### Competenze richieste

- ✓ Lettura di semplici mappe
- ✓ Riconoscimento di campioni (pattern)
- ✓ Logica
- ✓ Capacità di seguire le istruzioni

### Età

- ✓ A partire da 9 anni

### Materiale

L'insegnante deve avere:

- ✓ Un set di carte delle isole (le istruzioni vanno mantenute nascoste da chi tenta di disegnare la mappa!)  
Una copia delle carte delle isole (presenti a partire da pagina 104) debitamente ritagliate.  
Piegare le carte lungo la linea tratteggiata e incollatele in modo che da un lato sia presente il nome dell'isola e sull'altro le istruzioni.

Ogni studente deve avere:

- ✓ Il foglio di lavoro: Trova la via per le ricchezze dell'Isola del Tesoro (pagina 103)
- ✓ Penna o Matita

Ci sono anche attività opzionali di approfondimenti, per le quali ogni studente deve avere:

- ✓ Il foglio di lavoro: l'Isola del Tesoro (pagina 109)
- ✓ Il foglio di lavoro: Il misterioso gioco della moneta (pagina 114)

# L'Isola del Tesoro

---

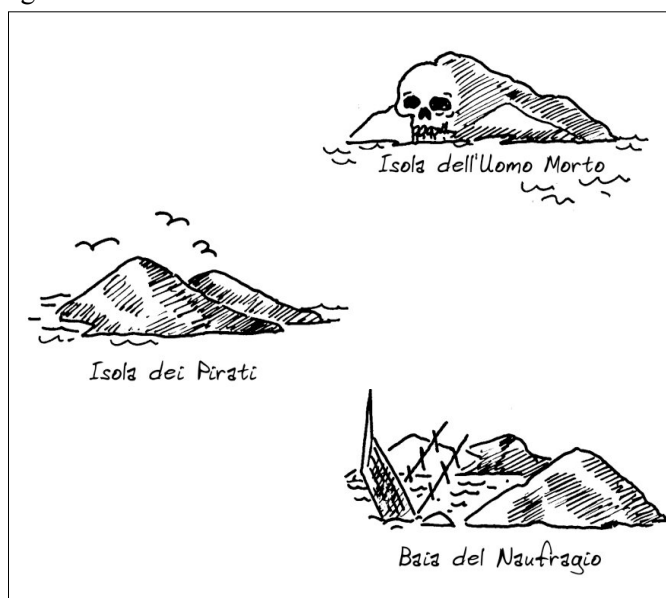
## Introduzione

Il vostro scopo è di trovare l'Isola del Tesoro. Navi pirata amiche veleggiano lungo un insieme di rotte fra le isole di questa parte del mondo, offrendo passaggi ai viaggiatori. Ogni isola ha due navi in partenza, A e B, sulle quali potete viaggiare. Dovete trovare la migliore via per l'Isola del Tesoro. Una volta giunti su di un'isola potete chiedere di ripartire o con la nave A o con la nave B (non entrambe). La persona all'isola vi dirà quindi su quale isola vi porterà la nave che avete scelto, ma i pirati non hanno una mappa con tutte le isole e le rotte. Dovete usare la vostra mappa per tenere traccia man mano di dove siete diretti e delle isole sulle quali siete già stati.

## Esempio pratico

(Nota: La mappa utilizzata è diversa da quella che gli studenti useranno nella attività)

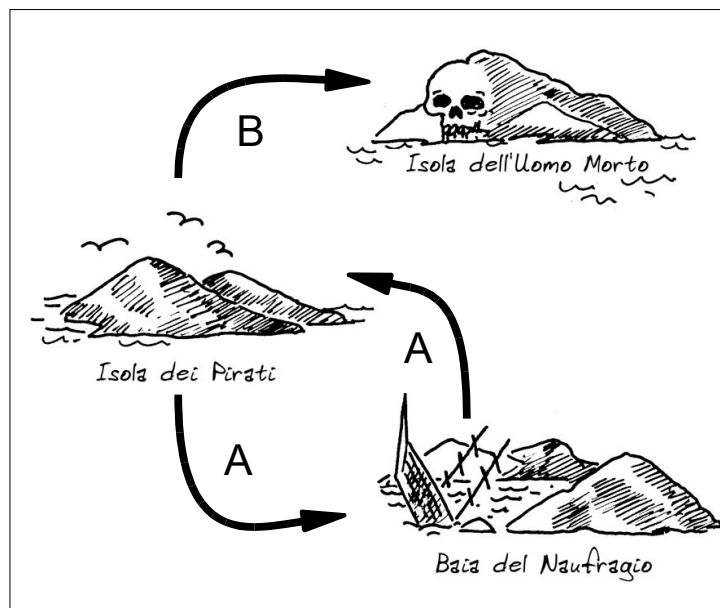
Usando un proiettore o la lavagna, mostrate o disegnate il diagramma delle tre isole come nella figura che segue:



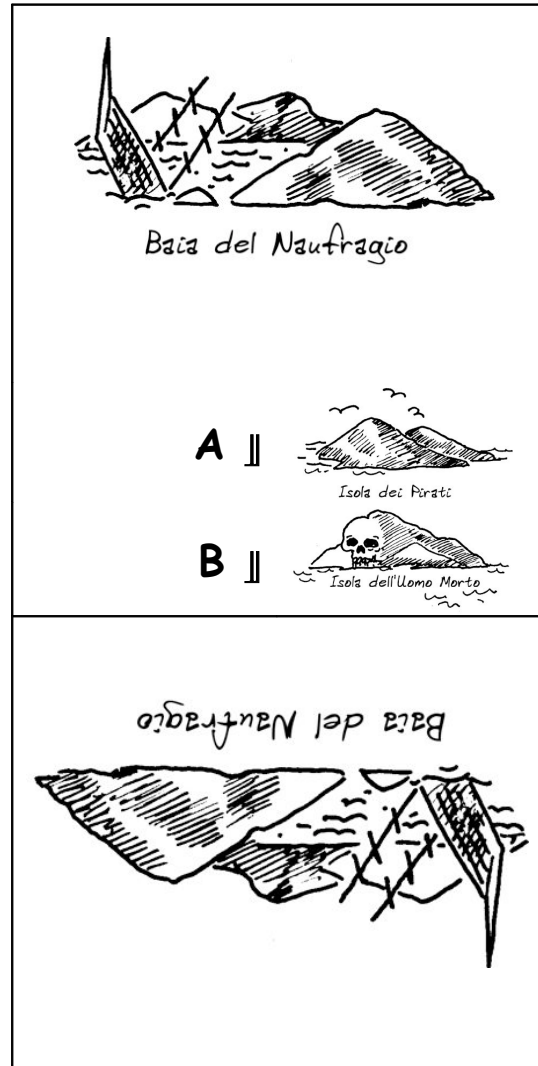
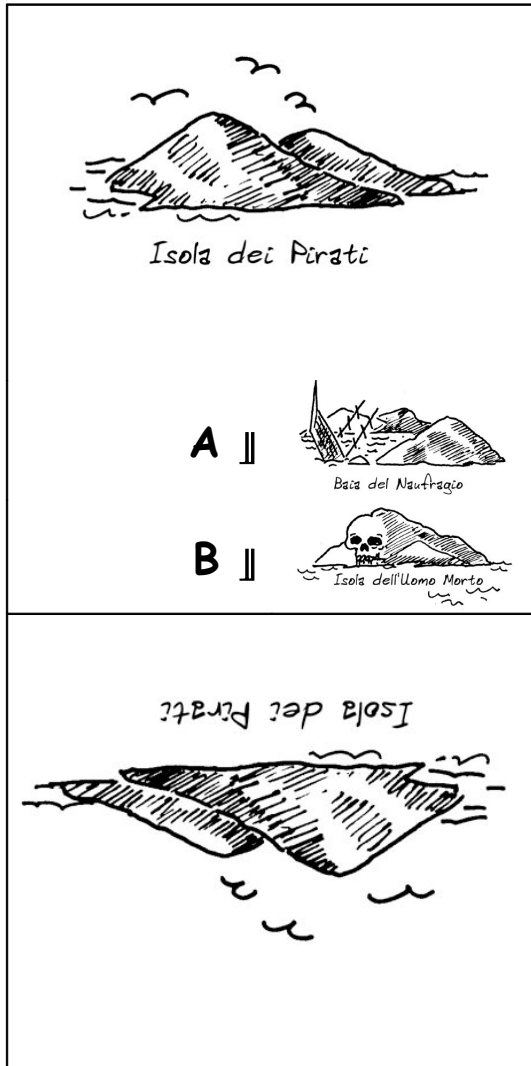
Copiate ritagliate e piegate le tre carte presenti nelle prossime due pagine. Scegliete tre studenti e fornite loro le carte in modo che ogni studente possa leggere solamente le regole della propria isola e che il resto della classe veda solo la corrispondenza fra i tre studenti e le tre isole. NB: I percorsi indicati in questo esempio saranno diversi da quelli dell'attività.

Partendo dall'Isola dei Pirati chiedete di usare la nave A. Lo studente con la carta dell'Isola dei Pirati vi dirà che la nave porta alla Baia del Naufragio. Segnate questo percorso sulla mappa. Alla Baia del Naufragio chiedete ancora di usare la nave A. Lo studente con la carta della Baia del Naufragio vi dirà che la nave vi riporta indietro all'Isola dei Pirati. Segnate anche questo percorso sulla mappa. Chiedete allora per la nave B che vi condurrà all'Isola dell'Uomo Morto. Segnate anche questo sulla mappa. A questo punto siete bloccati.

La vostra mappa alla fine dovrebbe apparire come questa:



## Carte per l'esempio pratico



## Carte per l'esempio pratico

---

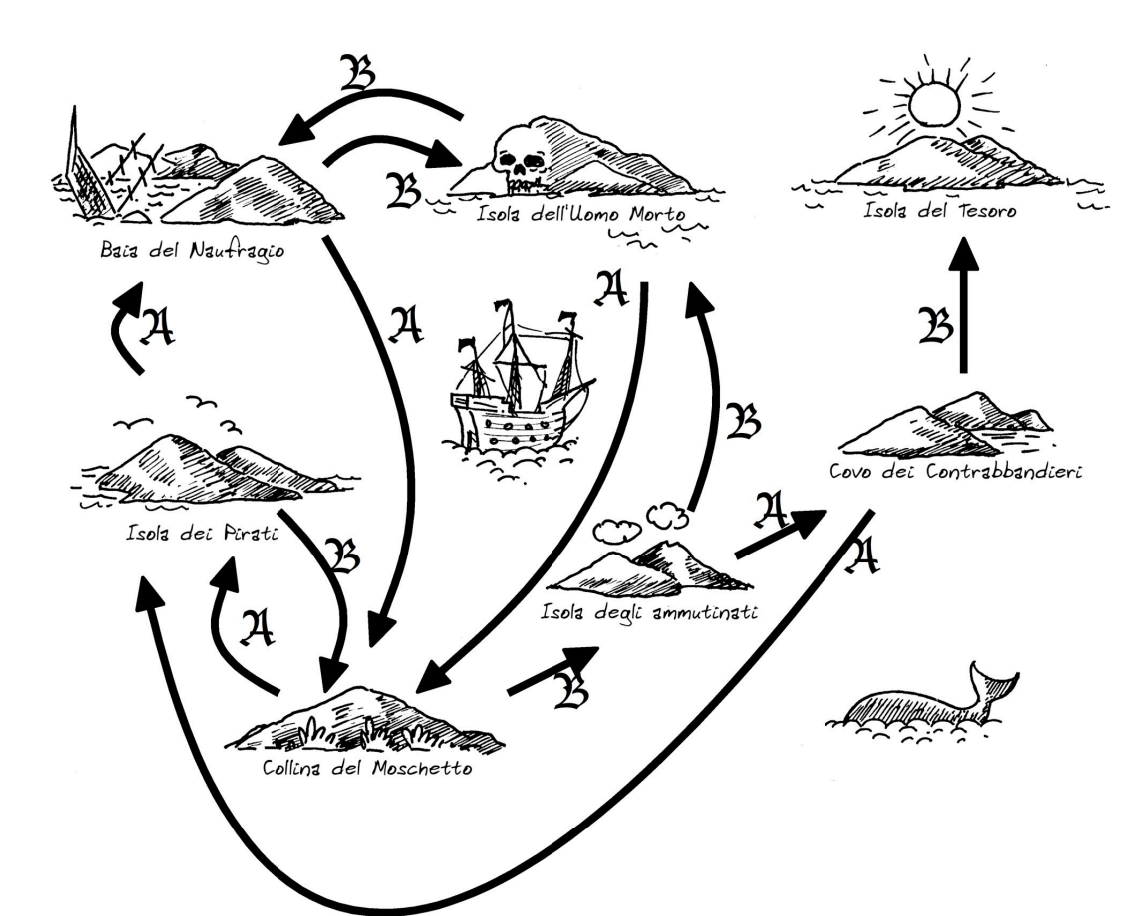


## Attività

Scegliete sette studenti che saranno le “isole”. Ognuno di questi sette studenti terrà la carta che rappresenta la propria isola, mantenendo le istruzioni segrete verso di sé in modo che il resto della classe non possa vederle. Posizionate questi studenti in modo casuale nella classe o nel giardino in modo che non siano troppo vicini fra loro. A tutti gli altri studenti date una copia della mappa bianca, con solo il disegno delle isole. Lo scopo dell'attività è di navigare per trovare una rotta che giunga dall'Isola dei Pirati all'Isola del Tesoro, indicandola sulla mappa a loro disposizione come spiegato nell'esempio pratico. (È consigliabile che gli studenti facciano il percorso uno alla volta perché non sentano in anticipo le rotte dei compagni).

Per chi finisce velocemente: cercare altre rotte alternative.

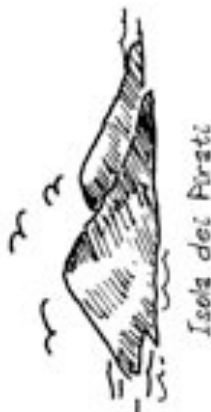
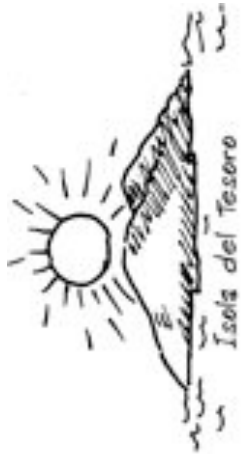
La mappa completa dovrebbe apparire così:



## Discussione

Qual è la rotta più veloce? Quale sarebbe una rotta molto lenta. Alcune rotte possono contenere cicli. Potete trovare esempi di rotte con cicli? (Per esempio, BBBABAB e BBBABBABAB, tutte portano all'Isola del Tesoro).

# Foglio di Lavoro: Trova la via per le ricchezze dell'Isola del Tesoro



# Pagina da fotocopiare: carte delle isole (1/4)



Isola dei Pirati

**A** ||



Baia del Naufragio

**B** ||



Collina del Moschetto



Baia del Naufragio

**A** ||



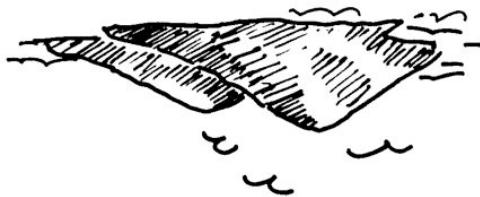
Collina del Moschetto

**B** ||



Isola dell'Uomo Morto

Isola dei Pirati

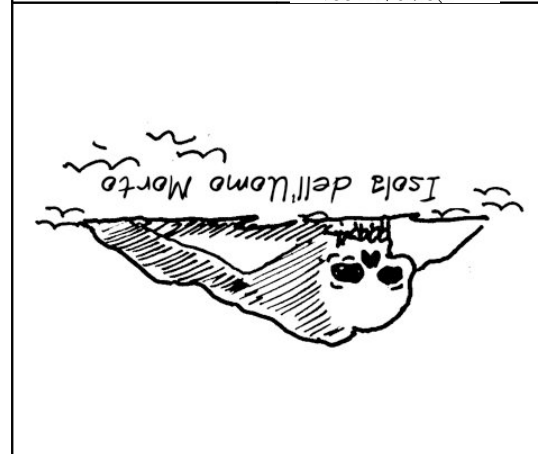
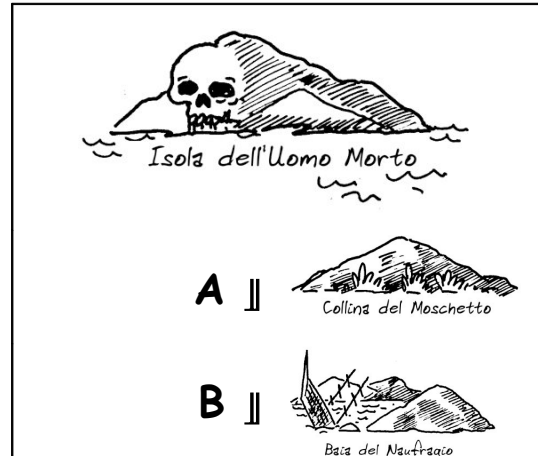
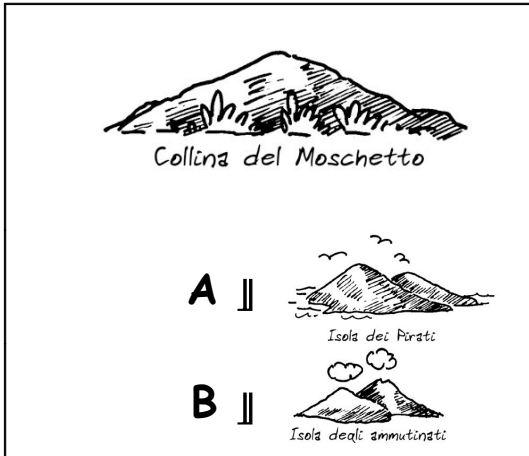


Baia del Naufragio





# Pagina da fotocopiare: carte delle isole (2/4)



# Pagina da fotocopiare: carte delle isole (3/4)


  
Isola degli ammutinati


**A** ||   
Covo dei Contrabbandieri


**B** ||   
Isola dell'Uomo Morto

*Isola degli ammutinati*




  
Covo dei Contrabbandieri

**A** ||   
Isola dei Pirati

**B** ||   
Isola del Tesoro

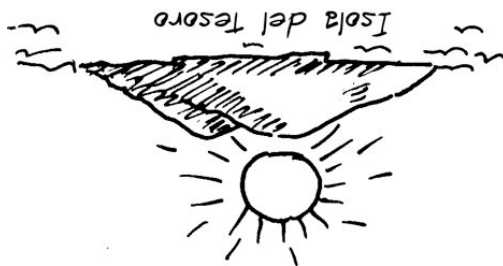
*Covo dei Contrabbandieri*



**Pagina da fotocopiare: carte delle isole (4/4)**



**Congratulazioni!**



## Automi a stati finiti

Un altro modo di disegnare una mappa è il seguente:

Le isole sono rappresentate da cerchietti con indicato un numero, l'isola di arrivo ha un doppio cerchietto. Quali rotte dobbiamo attraversare per arrivare all'isola finale negli esempi della figura qui sopra?

Note: Nella mappa (a) si arriva alla isola finale (la numero 2) solo se la sequenza contiene un numero dispari di A (per esempio AB, BABAA o AAABABA)

Nella mappa (b) si arriva all'isola finale solo con sequenze (anche vuote) di A e B dove i suoi simboli si alternano fra loro (AB, ABAB, ABABAB, ...).

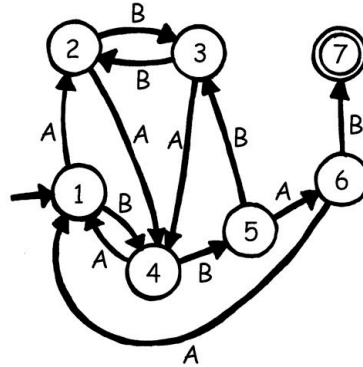
La mappa (c) richiede che la sequenza contenga almeno una B (le sole sequenze *non* accettate sono A, AA, AAA, AAAA, ...).

---

## Foglio di lavoro: l'isola del tesoro

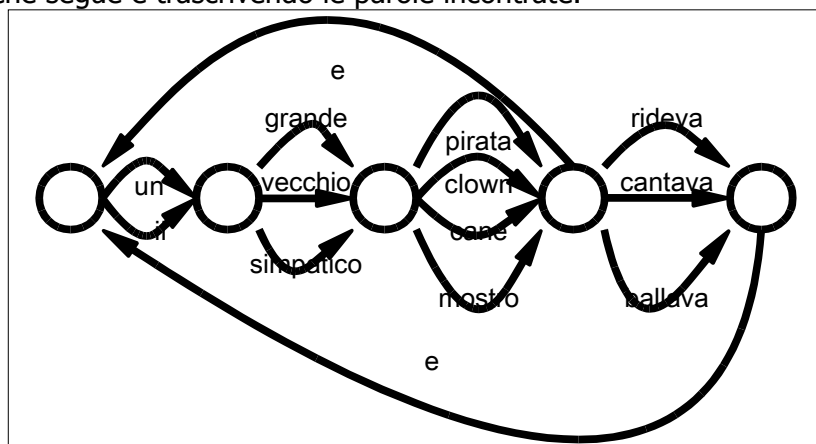
Siete capaci di sotterrare bene il vostro tesoro? Quanto potete rendere difficile trovare il tesoro? È ora di creare la vostra mappa!

1. Ecco una versione diversa della stessa idea di rappresentare una mappa. Questa è la mappa del precedente esercizio. Gli scienziati informatici usano questo metodo semplice e facile per disegnare le mappe utili a cercare particolari sequenze di simboli (dette pattern).



Disegnate la vostra mappa (simile a quella qui sopra ma con percorsi differenti) così da poter vedere chiaramente le rotte sulle quali viaggeranno le vostre navi pirata. Create quindi la vostra mappa bianca e le vostre carte delle isole. Qual è il percorso più efficiente per raggiungere la vostra isola del tesoro?

2. I vostri amici riescono a seguire bene la vostra mappa? Date loro una sequenza di A e di B e vedete se raggiungono l'isola corretta. Si possono fare tanti giochi e puzzle basati su questa idea di automa a stati finiti.
3. Ecco un modo per costruire frasi scegliendo casualmente percorsi attraverso la mappa che segue e trascrivendo le parole incontrate.



Ora provate a fare uno schema simile, magari potete anche costruire una storia divertente!

## Foglio di lavoro: il misterioso gioco della moneta.

Alcuni amici hanno scaricato da Internet un gioco nel quale un robot lancia in aria una moneta e occorre indovinare se viene testa o croce. All'inizio il gioco sembra facile, sembrano esserci 50 possibilità su 100 di vincere, o così pensano i ragazzi! Dopo un po' iniziano ad essere sospettosi. Sembra esserci una sequenza, un pattern ripetuto nel modo in cui la moneta appare. È un gioco truccato? Sicuramente no, pensano. Ma decidono ugualmente di investigare e Giovanni scrive i risultati di tutti i tiri della moneta (t=testa, c=croce):

```
t t c t t c t t t c c t t t t c c t c c c t t t t c t t t
c c c t t t c c c t t t t t c c t c c c c t c c t c c c
t t t c c t t t c t t t t t t t t c c t t t c c c c t t t
t t c c c c c c c
```

I nostri amici si devono ricredere, la sequenza non sembra proprio casuale. Potete trovare anche voi un pattern per prevedere i risultati dei prossimi tiri?

C'è una semplice mappa che descrive le sequenze dei tiri, provate a vedere se riuscite a trovarla (suggerimento: è composta da 4 isole).

---

# Cosa c'entra tutto questo?

---

Gli automi a stati finiti sono usati in informatica per riconoscere sequenze di caratteri o di eventi.

Un semplice esempio lo potete vedere, o meglio ascoltare, quando telefonando a una azienda o a un call center vi accoglie un messaggio registrato che dice “premi 1 per questo... premi 2 per quello... o premi 3 per parlare con un operatore umano”. Il tasto del telefono è l'input per una macchina a stati finiti. Il percorso nella macchina a stati finiti telefonica può essere semplice o complesso. Talvolta potete anche ritornare a un menù già visitato perché c'è un ciclo nell'automa a stati finiti. Quando questo accade è un errore di progetto del sistema ed è molto frustrante per chi chiama!

Un altro esempio lo potete trovare negli sportelli bancomat. Il programma della macchina vi conduce lungo una sequenza di eventi. All'interno di questo programma c'è un automa a stati finiti. Ogni tasto che premete porta l'automa in un nuovo stato (il nome informatico delle “isole”). Alcuni stati compiono azioni quali “fornisci 100 euro” o “stampa lo scontrino” oppure ancora “espelli la tessera”.

Alcuni programmi veramente trattano frasi in italiano o in inglese usando mappe come quella di pagina 109. Possono essere usati per generare frasi e per elaborare frasi in ingresso. Negli anni 1960 uno scienziato informatico scrisse un programma famoso chiamato “Eliza” (dal nome di Eliza Dolittle, la protagonista di *My Fair Lady*) che era in grado di sostenere conversazioni con le persone. Il programma pretendeva di essere uno psicologo e faceva domande come “Mi dica qualcosa della sua famiglia”, e anche “Suvvia, vada avanti”. Anche se il programma non “capiva” nulla era sufficientemente credibile, e gli utenti umani sufficientemente creduloni, che qualcuno era veramente convinto di parlare con un vero psicologo.

Anche se i computer non sono bravi a comprendere il linguaggio naturale, possono elaborare rapidamente linguaggi artificiali. Un tipo importante di linguaggio artificiale sono i linguaggi di programmazione. I computer usano macchine a stati finiti per leggere i programmi e trasformarli nella forma di istruzioni elementari del computer, che possono essere “eseguite” direttamente dal computer stesso.



(Grr! Questa non può essere una mappa del tesoro! Dove è la X?)

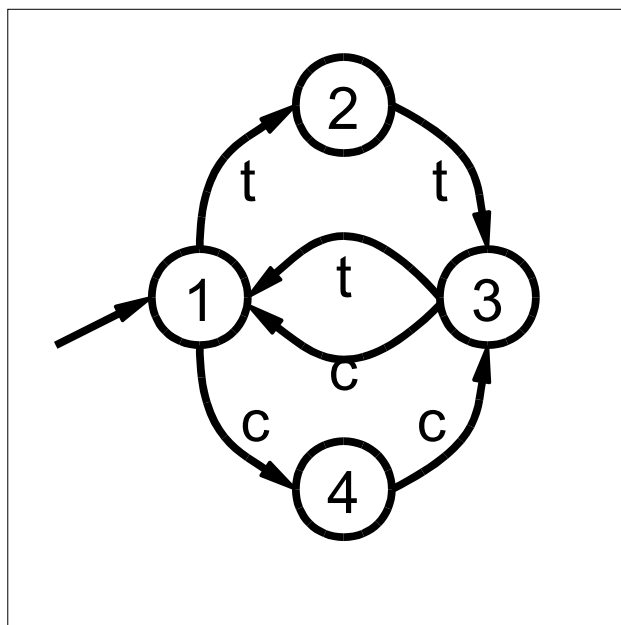


# Soluzioni e suggerimenti

---

Il misterioso gioco della moneta (pagina 114)

Il misterioso gioco della moneta usa questa mappa per i risultati dei lanci delle monete:



Se lo seguirete, scoprirete che i primi due tiri ogni tre danno sempre lo stesso risultato.

# Attività 12

---

## **Gli ordini di marcia—*I linguaggi di programmazione***

### **Sommario**

I computer vengono programmati mediante un “linguaggio” composto da un numero limitato di ordini che possono essere eseguiti. Uno dei lati più frustranti dello scrivere programmi è che i computer obbediscono sempre alla lettera agli ordini impartiti, anche quando per colpa di “malintesi” questi producono un risultato chiaramente diverso da quello atteso. Questa attività fornisce agli studenti alcune esperienze su questo aspetto della programmazione.

### **Competenze richieste**

- ✓ Saper dare e seguire sequenze di istruzioni

### **Età**

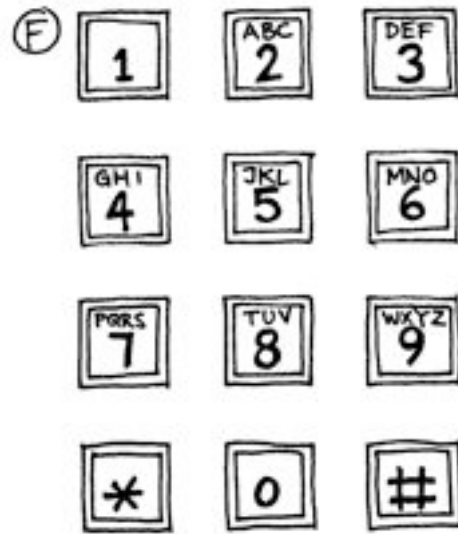
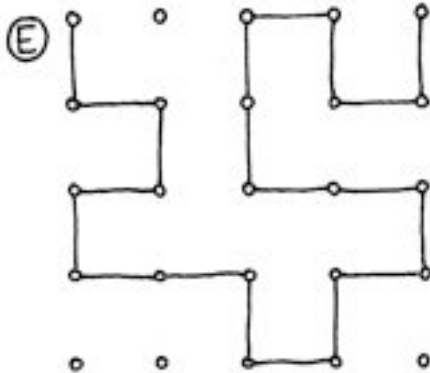
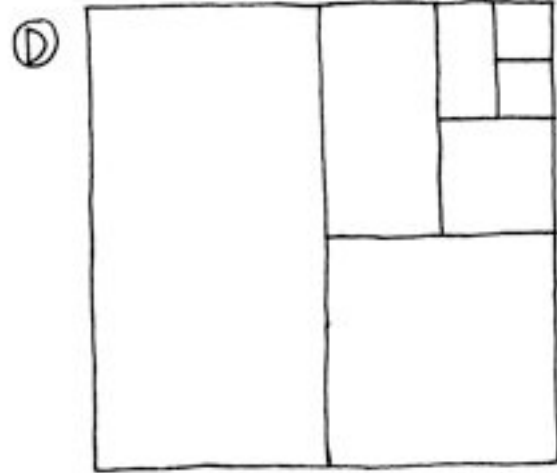
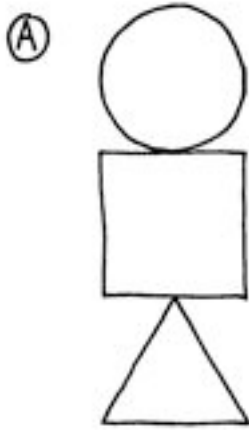
- ✓ A partire da 7 anni

### **Materiale**

- ✓ Carte con figure come quelle mostrate nella pagina seguente.

Ogni studente deve avere:

- ✓ Matita, carta e righello.
-



# Ordini di marcia

---

## Introduzione

Discutere se sarebbe bene che tutte le persone seguissero esattamente le istruzioni. Per esempio cosa succederebbe se indicando una porta chiusa qualcuno desse l'ordine "Passa attraverso quella porta"?

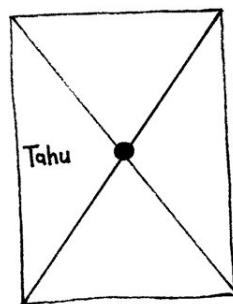
I computer lavorano seguendo una lista di istruzioni, e fanno esattamente quello che le istruzioni dicono, compreso quando le istruzioni non hanno alcun senso!

## Esempio pratico

Chiedete agli studenti di disegnare su un foglio bianco una figura seguendo attentamente le istruzioni seguenti:

1. Disegna un punto nel centro del foglio.
2. A partire dall'angolo in alto a sinistra traccia una riga dritta che passi attraverso il punto disegnato e termini all'angolo in basso a destra.
3. A partire dall'angolo in basso a sinistra traccia una riga dritta che passi attraverso il punto disegnato e termini all'angolo in alto a destra.
4. Scrivi ora il tuo nome nel triangolo posto al centro del lato sinistro del foglio.

Il risultato dovrebbe apparire simile a questo:



## **Attività**

Scegliete uno studente e dategli una immagine (per esempio uno degli esempi di pagina 115). L'immagine deve rimanere nascosta al resto della classe. Lo studente deve descrivere l'immagine al resto della classe che devono tentare di riprodurla seguendo esclusivamente le istruzioni impartite. Gli altri studenti possono chiedere di precisare le istruzioni qualora non risultassero chiare. L'obiettivo dell'esperimento è di vedere quanto l'esercizio venga completato in modo veloce e accurato.

Ripetete l'esperimento, questa volta non consentendo più domande da parte della classe. È meglio scegliere immagini semplici perché gli studenti molto facilmente potranno perdersi.

Ora ripetete l'esperimento con lo studente che fornisce le informazioni nascosto dietro uno schermo in modo che non ci sia alcuna forma di comunicazione se non la sequenza delle istruzioni.

Fate notare che è proprio questa la forma di comunicazione che i programmatori hanno col computer quando scrivono programmi. Essi infatti scrivono sequenze di istruzioni per il computer e non possono vedere l'effetto se non più tardi, quando il programma viene eseguito.

Ora potete far fare semplici disegni agli studenti e chiedere che scrivano la sequenza delle istruzioni per realizzare il disegno. Possono fare l'esercizio a coppie (scambiandosi le istruzioni per vedere di ottenere nuovamente il disegno del collega) o con tutta la classe.

## **Variazioni.**

1. Scrivere le istruzioni per realizzare una freccetta o un aeroplanino di carta.
2. Scrivere le istruzioni per raggiungere un punto misterioso nella scuola o nel cortile con istruzioni come “avanza di  $x$  metri”, “gira a sinistra” (90 gradi), “gira a destra” (90 gradi). Gli studenti dovrebbero provare le istruzioni più e più volte fino a che non riescono ad ottenere il risultato voluto.
3. Mosca cieca. Bendate uno studente e chiedete agli altri di darli le istruzioni per muoversi nella classe e compiere alcune azioni.

# Cosa c'entra tutto questo?

---

I computer funzionano seguendo una lista di istruzioni, chiamato *programma*, che è stato scritto per svolgere un particolare compito. I programmi sono scritti con linguaggi che sono stati creati appositamente, con un insieme limitato di istruzioni, per dire al computer cosa fare. Alcuni linguaggi sono più adatti ad alcuni compiti rispetto ad altri.

Indipendentemente dal linguaggio di programmazione usato, i programmatori devono stare molto attenti a specificare esattamente cosa vogliono che il computer faccia. Differentemente dagli esseri umani, un computer esegue le istruzioni alla lettera anche queste sono palesemente errate, magari ridicole.

È importante allora che i programmi siano scritti bene. Un piccolo errore può causare molti problemi. Immaginare le conseguenze che un errore di un programma può avere nel computer che controlla il lancio di uno space shuttle, o nel computer di controllo di una centrale nucleare o in quello che accende i segnali di controllo per i treni! Gli errori dei programmi sono di solito chiamati “bug” cioè scarafaggi in onore (si fa per dire) di un “bug” che fu rimosso da un relais elettromagnetico di uno dei primi computer all'inizio degli anni 1940. Lo scarafaggio aveva causato il malfunzionamento del computer, e ancora oggi la ricerca degli errori nei programmi viene chiamata “debugging” cioè disinfezione.



Più un programma è complesso, più errori può contenere. Negli USA la possibile presenza di errori nei programmi è diventato un problema enorme nel progetto del Ministero della Difesa chiamato “Guerre Stellari”. Il programma prevede la creazione di un sistema controllato da computer che crei uno scudo impenetrabile per difendersi da attacchi nucleari. Alcuni scienziati informatici hanno sostenuto che il progetto potrebbe non funzionare per la complessità e quindi la conseguente inaffidabilità dei programmi necessari. I programmi devono essere controllati accuratamente per cercare più bug che si può, ma non si può aver alcuna certezza di averli trovati tutti. E non è certo il caso di sparare missili nucleari per vedere se funziona davvero oppure no!