



**INFORMATIK-BIBER SCHWEIZ  
CASTOR INFORMATIQUE SUISSE  
CASTORO INFORMATICO SVIZZERA**

## Quesiti e soluzioni 2017 Tutte le Categorie

<http://www.castoro-informatico.ch/>

A cura di:

Andrea Adamoli, Christian Datzko, Hanspeter Erni

010100110101011001001001  
010000010010110101010011  
010100110100100101000101  
001011010101001101010011  
010010010100100100100001

# SSI

[www.svia-ssie-ssii.ch](http://www.svia-ssie-ssii.ch)  
schweizerischerverein für informatik in d  
erausbildung // société suisse pour l'infor  
matique dans l'enseignement // società sviz  
zera per l'informatica nell'insegnamento





# Hanno collaborato al Castoro Informatico 2017

Andrea Adamoli, Christian Datzko, Susanne Datzko, Olivier Ens, Hanspeter Erni, Martin Gugger, Per Matzinger, Carla Monaco, Nicole Müller, Gabriel Parriaux, Jean-Philippe Pellet, Julien Ragot, Silvan Stöckli, Beat Trachsler.

Un particolare ringraziamento va a:

Juraj Hromkovič, Giovanni Serafini, Urs Hauser, Regula Lacher, Ivana Kosírová: ETHZ

Valentina Dagiene: Bebras.org

Hans-Werner Hein, Wolfgang Pohl: Bundesweite Informatikwettbewerbe (BWINF), Germania

Anna Morpurgo, Violetta Lonati, Mattia Monga: Italia

Gerald Futschek, Wilfried Baumann: Austrian Computer Society, Austria

Zsuzsa Pluhár: ELTE Informatikai Kar, Ungheria

Eljakim Schrijvers, Daphne Blokhuis: Eljakim Information Technology bv, Paesi Bassi

Roman Hartmann: hartmannGestaltung (Flyer Castoro Informatico Svizzera)

Christoph Frei: Chragokyberneticks (Logo Castoro Informatico Svizzera)

Pamela Aeschlimann, Andreas Hieber, Aram Loosmann, Daniel Vuille, Peter Zurflüh: Lernetz.ch (pagina web)

Andrea Leu, Maggie Winter, Brigitte Maurer: Senarclens Leu + Partner

L'edizione dei quesiti in lingua tedesca è stata utilizzata anche in Germania e in Austria.

La traduzione francese è stata curata da Nicole Müller mentre quella italiana da Andrea Adamoli.



**INFORMATIK-BIBER SCHWEIZ**  
**CASTOR INFORMATIQUE SUISSE**  
**CASTORO INFORMATICO SVIZZERA**

Il Castoro Informatico 2017 è stato organizzato dalla Società Svizzera per l'Informatica nell'Insegnamento SSII. Il Castoro Informatico è un progetto della SSII con il prezioso sostegno della fondazione Hasler.

## HASLERSTIFTUNG

Nota: Tutti i link sono stati verificati l'01.11.2017. Questo quaderno è stato creato il 9 ottobre 2019 col sistema per la preparazione di testi L<sup>A</sup>T<sub>E</sub>X.



I quesiti sono distribuiti con Licenza Creative Commons Attribuzione – Non commerciale – Condividi allo stesso modo 4.0 Internazionale. Gli autori sono elencati a pagina 99.



## Premessa

Il concorso del «Castoro Informatico», presente già da diversi anni in molti paesi europei, ha l'obiettivo di destare l'interesse per l'informatica nei bambini e nei ragazzi. In Svizzera il concorso è organizzato in tedesco, francese e italiano dalla Società Svizzera per l'Informatica nell'Insegnamento (SSII), con il sostegno della fondazione Hasler nell'ambito del programma di promozione «FIT in IT».

Il Castoro Informatico è il partner svizzero del Concorso «Bebras International Contest on Informatics and Computer Fluency» (<http://www.bebas.org/>), situato in Lituania.

Il concorso si è tenuto per la prima volta in Svizzera nel 2010. Nel 2012 l'offerta è stata ampliata con la categoria del «Piccolo Castoro» (3<sup>o</sup> e 4<sup>o</sup> anno scolastico).

Il «Castoro Informatico» incoraggia gli alunni ad approfondire la conoscenza dell'Informatica: esso vuole destare interesse per la materia e contribuire a eliminare le paure che sorgono nei suoi confronti. Il concorso non richiede alcuna conoscenza informatica pregressa, se non la capacità di «navigare» in Internet poiché viene svolto online. Per rispondere alle domande sono necessari sia un pensiero logico e strutturato che la fantasia. I quesiti sono pensati in modo da incoraggiare l'utilizzo dell'informatica anche al di fuori del concorso.

Nel 2017 il Castoro Informatico della Svizzera è stato proposto a cinque differenti categorie d'età, suddivise in base all'anno scolastico:

- 3<sup>o</sup> e 4<sup>o</sup> anno scolastico («Piccolo Castoro»)
- 5<sup>o</sup> e 6<sup>o</sup> anno scolastico
- 7<sup>o</sup> e 8<sup>o</sup> anno scolastico
- 9<sup>o</sup> e 10<sup>o</sup> anno scolastico
- 11<sup>o</sup> al 13<sup>o</sup> anno scolastico

Gli alunni iscritti al 3<sup>o</sup> e 4<sup>o</sup> anno scolastico hanno dovuto risolvere 9 quesiti (3 facili, 3 medi e 3 difficili).

A ogni altra categoria d'età sono stati assegnati 15 quesiti da risolvere, suddivisi in gruppi di cinque in base a tre livelli di difficoltà: facile, medio e difficile. Per ogni risposta corretta sono stati assegnati dei punti, mentre per ogni risposta sbagliata sono stati detratti. In caso di mancata risposta il punteggio è rimasto inalterato. Il numero di punti assegnati o detratti dipende dal grado di difficoltà del quesito:

	Facile	Medio	Difficile
Risposta corretta	6 punti	9 punti	12 punti
Risposta sbagliata	-2 punti	-3 punti	-4 punti

Il sistema internazionale utilizzato per l'assegnazione dei punti limita l'eventualità che il partecipante possa indovinare la risposta corretta.

Ogni partecipante aveva un punteggio iniziale di 45 punti (Piccolo Castoro 27).

Il punteggio massimo totalizzabile era pari a 180 punti (Piccolo castoro 108), mentre quello minimo era di 0 punti.

In molti quesiti le risposte possibili sono state distribuite sullo schermo con una sequenza casuale. Lo stesso quesito è stato proposto in più categorie d'età.



---

**Per ulteriori informazioni:**


SVIA-SSIE-SSII Società Svizzera per l'Informatica nell'Insegnamento

Castoro Informatico

Andrea Adamoli

[castoro@castoro-informatico.ch](mailto:castoro@castoro-informatico.ch)

<http://www.castoro-informatico.ch/>

 <https://www.facebook.com/informatikbiberch>



# Indice

Hanno collaborato al Castoro Informatico 2017	i
Premessa	ii
1. Parcheggio	1
2. Casetta per canarini	3
3. Trova la strada	5
4. Cancelli binari	7
5. Servizio di messaggistica	9
6. Caccia alla fragola	11
7. Scambio di cani	13
8. Il castoro infortunato	15
9. Abbattere muri	19
10.5 bastoncini	23
11. Esibizione di ballo	25
12. Il nome giapponese	29
13. Un piccolo programma	31
14. Gioielli	33
15. Il giornale della scuola	35
16. Honomakato	39
17. L'arte del bastone giapponese	43
18. La marmellata della nonna	45
19. La città delle rotonde	47
20. La pizzeria dei castori	49
21. Ordinazione segreta	51
22. Gioco di monete	53
23. Succhi di frutta	55
24. Intrusione al museo	59



---

25. Giochi di luce	61
26. Sostituzioni	63
27. Esci dal labirinto!	65
28. Impianto di irrigazione	67
29. Una nuova canzone	69
30. Pista per biglie	71
31. I tunnel della diga	75
32. Aiuta l'Arabot	79
33. Gioco con stuzzicadenti	81
34. La distanza tra parole	83
35. Download contemporanei	85
36. Raccolta di punti	87
37. Metodo dei quarti	91
38. Scorciatoia o strada più lunga?	93
39. Display digitale	95
40. Suddivisione del codice	97
A. Autori dei quesiti	99
B. Sponsoring: concorso 2017	100
C. Ulteriori offerte	102

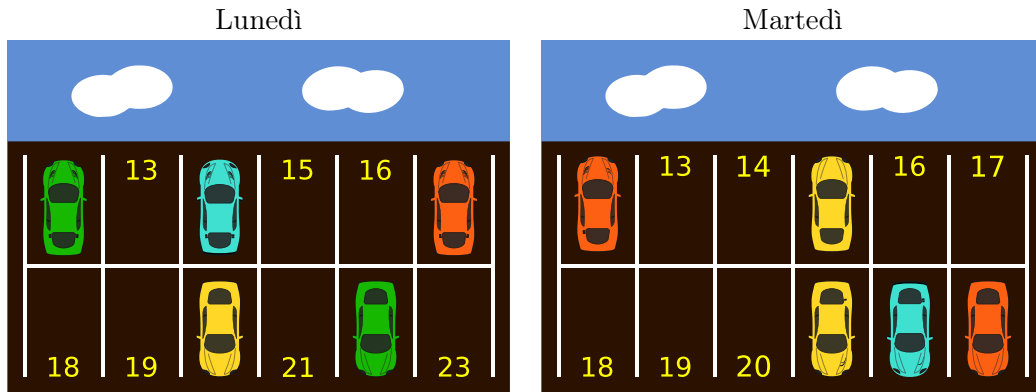






# 1. Parcheggio

Nella piazza dei castori ci sono parcheggi per 12 automobili. Ogni parcheggio è identificato con un numero. Le immagini qui sotto indicano quali parcheggi sono stati occupati Lunedì e quali Martedì.



Quanti parcheggi erano liberi sia Lunedì, sia Martedì?

- A) 3
- B) 4
- C) 5
- D) 6

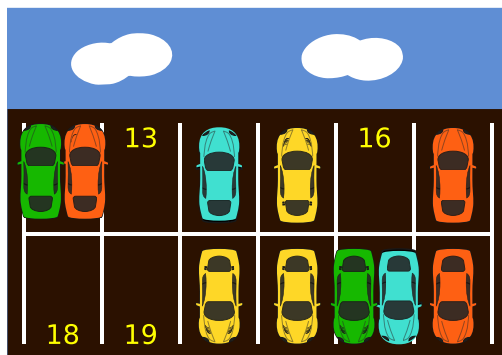


## Soluzione

4 parcheggi erano liberi sia Lunedì, sia Martedì.

Se sovrapponiamo le auto parcheggiate Lunedì e Martedì in un'unica immagine, possiamo vedere facilmente quali posti sono rimasti liberi in entrambi i giorni. Gli spazi rimasti vuoti possono dunque essere contati. In totale, 4 posti (13, 16, 18, 19) non sono mai stati occupati.

Sovrapposizione dei parcheggi di Lunedì e Martedì:



## Questa è l'informatica!

Tutti i dati possono essere codificati come una sequenza di *zero* e di *uno*. Ogni zero ed ogni uno viene chiamato bit. Ogni sequenza viene detta codice binario.

In questo compito, possiamo identificare un'auto in un parcheggio con un *uno* (1) e un posto libero con uno *zero* (0). In tal modo possiamo modellare l'occupazione dei parcheggi con una sequenza di bit. Lunedì è rappresentato con 101001001010, mentre Martedì con 100100000111. Con l'aiuto di un operatore logico OR (o, oppure), rappresentiamo i parcheggi occupati Lunedì o Martedì con un 1. Allineando le due sequenze in colonna, possiamo quindi calcolare la risposta:

$$\begin{array}{r}
 101001001010 \\
 \text{o} \\
 100100000111 \\
 = \\
 101101001111
 \end{array}$$

Quest'ultima sequenza di bit contiene 4 zero, dunque 4 posti sono rimasti liberi in entrambi i giorni.

## Parole chiave e siti web

Bit, codice binario, OR logico, operatore logico

- [https://it.wikipedia.org/wiki/Disgiunzione\\_logica](https://it.wikipedia.org/wiki/Disgiunzione_logica)



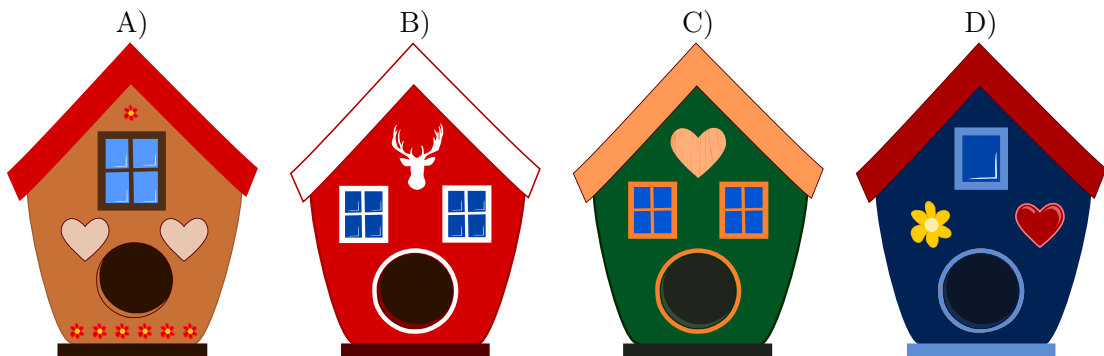
## 2. Casetta per canarini

La mamma castoro vuole regalare una casetta per canarini alla propria bambina. Chiede quindi alla bambina quale modello di casetta preferisca. La bambina risponde:

*“Desidero una casetta per canarini con 2 finestre e un cuore”*

La mamma va allora nel negozio di animali per comprarle la casetta.

*Quale è la casetta che la mamma castoro compra per la propria bambina?*





## Soluzione

La risposta corretta è C), poiché la casetta ha 2 finestre e un cuore.

La risposta A) è sbagliata poiché la casetta ha una sola finestra e 2 cuori.

La risposta B) è sbagliata poiché la casetta non ha cuori.

La risposta D) è sbagliata poiché la casetta ha una sola finestra.

## Questa è l'informatica!

La difficoltà di questo compito consiste nel soddisfare i desideri della bambina con una delle cassette presenti nel negozio. Il colore o altre caratteristiche della casetta non sono importanti e bisogna pertanto ignorarle.

Questa visione "limitata" viene chiamata astrazione: solo determinate proprietà vengono tenute in considerazione. In informatica, questo è spesso necessario.

## Parole chiave e siti web

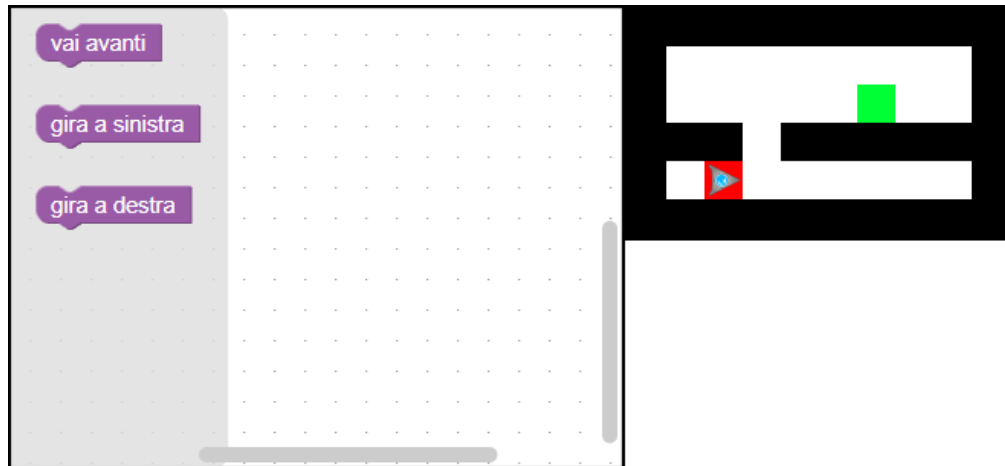
Modello (pattern), riconoscimento di modelli (pattern recognition), proprietà, astrazione

- [https://it.wikipedia.org/wiki/Riconoscimento\\_di\\_pattern](https://it.wikipedia.org/wiki/Riconoscimento_di_pattern)
- [https://it.wikipedia.org/wiki/Astrazione\\_\(informatica\)](https://it.wikipedia.org/wiki/Astrazione_(informatica))



### 3. Trova la strada

Un robot a forma di triangolo si trova nella posizione di partenza segnata in rosso e deve arrivare al traguardo colorato di verde.



*Programma il robot spostando le istruzioni nella giusta sequenza all'interno del pannello di programmazione.*



## Soluzione

Una possibile soluzione è:



Per raggiungere il traguardo dalla posizione di partenza, il robot deve dapprima compiere un passo in avanti, in modo da trovarsi vicino alla buca della parete di sinistra. Se poi gira a sinistra e compie due passi in avanti, supererà tale buca. Il robot potrà poi girare a destra e raggiungere il traguardo con ulteriori 3 passi in avanti.

## Questa è l'informatica!

Nella robotica “mobile” la navigazione è un problema generale. Trovare la soluzione per uscire da un labirinto non è molto comune, ma richiede grosse abilità di pensiero computazionale.

I labirinti possono avere diverse caratteristiche: essi possono contenere o no dei cicli (loop), oppure possono essere o no basati su delle griglie.

Nel nostro caso, il robot non possiede sensori, e dunque è cieco. Esso non può quindi trovare la strada da solo, ma deve affidarsi alle tue istruzioni per superare il buco nella parete e arrivare al traguardo. Devi quindi programmare il robot!

## Parole chiave e siti web

(sequenze di) istruzioni, programmazione

- <https://en.wikipedia.org/wiki/Blockly>
- <https://scratch.mit.edu/>
- <http://code.org/>
- <http://insegnamento.adamoli.ch/>



## 4. Cancelli binari

I castori si rendono visita l'un l'altro molto spesso. ... a volte, però, non c'è nessuno a casa. In tal caso i padroni di casa lasciano un messaggio al cancello in pietra del giardino, per comunicare quando torneranno. Per fare questo, possono infilare fino a 3 bastoni nei fori opposti delle lastre in pietra.

I castori hanno concordato i seguenti messaggi:



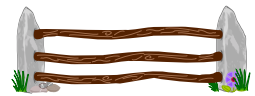
Siamo a casa, entra.



Ritourneremo a mezzo-giorno.



Purtroppo torneremo solo questa sera.



Stiamo visitando qualcuno e torneremo solo a tarda notte.

I castori potrebbero concordare ancora altri messaggi, senza dover aggiungere dei bastoni o dei fori nelle pietre.

*Quanti messaggi diversi si potrebbero concordare in totale, contando anche i 4 mostrati sopra?*



## Soluzione

In totale, possono essere concordati 8 messaggi diversi.



## Questa è l'informatica!

In questo compito, i castori utilizzano un sistema binario con tre posizioni. Le informazioni sono codificate attraverso le coppie opposte di fori delle pietre. Ogni coppia di fori può avere 2 stati: “bastone infilato”, oppure, “bastone non infilato”. La posizione di queste coppie è importante per poter distinguere i diversi messaggi (“notazione posizionale”, come ad esempio per i numeri). Il numero totale di messaggi corrisponde dunque al numero di stati delle coppie di fori (2) elevato al numero di coppie di fori (3), cioè:  $2^3 = 2 \cdot 2 \cdot 2 = 8$ .

I castori conoscono il significato dei vari messaggi, ma se ci fosse un errore nell'interpretarli non ci sarebbero grossi problemi. In informatica, al contrario, è importante che ogni messaggio (composto da cifre binarie) che viaggia sulla rete globale, sia correttamente identificato senza errori.

Grosse organizzazioni si occupano della standardizzazione dei protocolli di comunicazioni basati sui codici binari e della certificazioni dei prodotti che li sfruttano. Nei comitati di esperti internazionali, vengono discussi e decisi significato e forma dei vari protocolli. I sistemi di comunicazione basati sulla segnaletica, ad esempio, sono dichiarati legge dai parlamenti di molti stati.

## Parole chiave e siti web

codifica, codice binario, standard

- [https://it.wikipedia.org/wiki/Binary-coded\\_decimal](https://it.wikipedia.org/wiki/Binary-coded_decimal)
- [https://it.wikipedia.org/wiki/Notazione\\_posizionale](https://it.wikipedia.org/wiki/Notazione_posizionale)





## 5. Servizio di messaggistica

Con l'aiuto dei castori, Violetta desidera spedire un messaggio a Leonardo. Per farlo divide il messaggio in più parti, ognuna composta da 3 lettere scritte su un bigliettino, e poi consegna uno di questi bigliettini ad ogni castoro.

Violetta sa che durante il trasporto i castori possono essere rallentati da degli ostacoli e quindi potrebbero arrivare in ordine sparso. Per questo numera i bigliettini prima di consegnarli ai castori. Una volta ricevuti, Leonardo deve riordinare nella giusta sequenza i bigliettini prima di poter leggere il messaggio completo.

Per poter inviare il messaggio FESTAOGGI, Violetta prepara 3 bigliettini come segue:



Leonardo riceve ora questa sequenza di bigliettini:



Qual è il messaggio originale?

- A) UILLATAQPORAPA
- B) LLATAQUILAPAUIL
- C) PORTAQUILAPALLA
- D) APALLAPORTAQUIL



## Soluzione

La risposta giusta è C) PORTAQUILAPALLA.

Ordinando nel modo giusto i bigliettini, si ottiene il seguente messaggio:



## Questa è l'informatica!

Quando si inviano dati (ad esempio e-mail, foto o video) attraverso la rete internet, essi vengono spezzettati in piccoli pacchetti (pacchetti TCP/IP). Ogni pacchetto ha una grandezza massima di 65'536 caratteri ( $2^{16} = 65'536$  caratteri = 64 KB).

Questi pacchetti vengono poi inviati assieme a delle informazioni aggiuntive (mittente, destinatario, numero di sequenza del pacchetto, ...) attraverso svariati router. Tutte queste informazioni aggiuntive fanno in modo che il destinatario sia in grado di ricostruire la sequenza ordinata dei pacchetti e quindi l'intero messaggio originale.

Con il nuovo protocollo IPv6 è possibile inviare pacchetti di dimensioni maggiori.

## Parole chiave e siti web

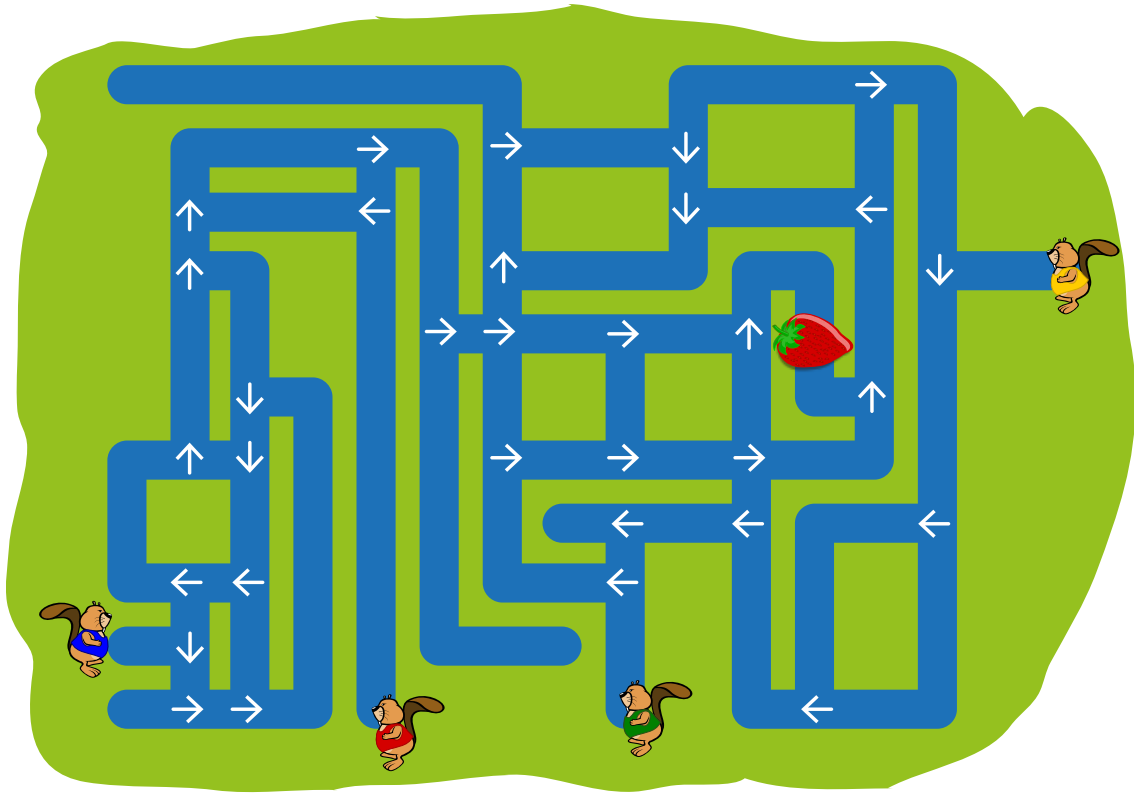
protocollo internet (IP), pacchetto di dati

- [https://it.wikipedia.org/wiki/Pacchetto\\_\(reti\)](https://it.wikipedia.org/wiki/Pacchetto_(reti))
- [https://en.wikipedia.org/wiki/IP\\_fragmentation](https://en.wikipedia.org/wiki/IP_fragmentation)



## 6. Caccia alla fragola

Quattro castori iniziano a nuotare da quattro posizioni iniziali diverse, seguendo la direzione delle frecce ad ogni incrocio.



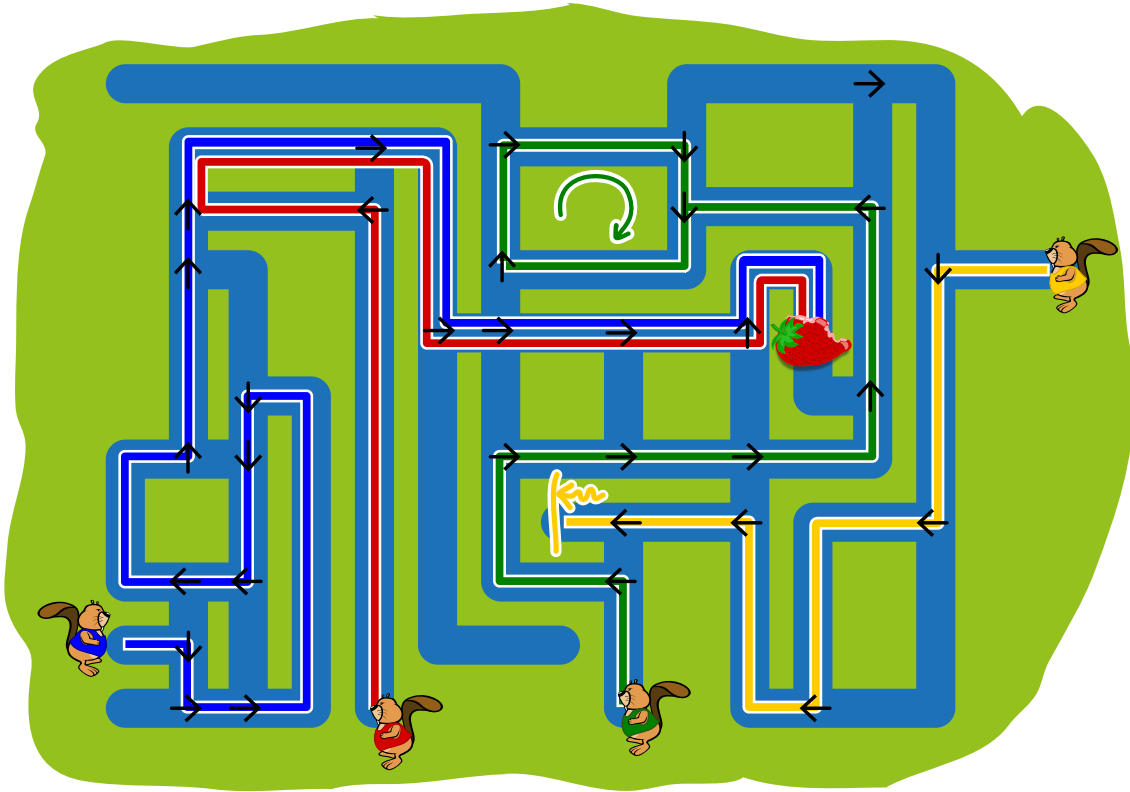
*Quanti castori arriveranno fino alla fragola?*

- A) 0
- B) 1
- C) 2
- D) 3
- E) 4



## Soluzione

Solo due dei quattro castori arrivano fino alla fragola e più precisamente i due più a sinistra:



Il terzo castoro da sinistra alla fine si troverà a nuotare in cerchio, mentre il castoro a destra giungerà in un vicolo cieco, dal quale non potrà uscire.

## Questa è l'informatica!

Nel sistema mostrato dalla figura possiamo individuare due elementi: i canali (attraverso i quali nuotano i castori) e gli incroci (nei quali le frecce indicano che direzione bisogna prendere). In informatica sistemi simili sono rappresentati attraverso dei *grafi*, dove gli *archi* rappresentano i canali e i *nodi* gli incroci. Archi e nodi possono contenere delle informazioni: nel nostro caso i nodi indicano quali arco deve essere seguito.

I grafi possono essere utilizzati in molte altre situazioni, ad esempio per la programmazione dei computer. I nodi possono essere associati alle istruzioni e seguendo gli archi che le collegano possiamo sapere quale istruzione verrà eseguita subito dopo.

## Parole chiave e siti web

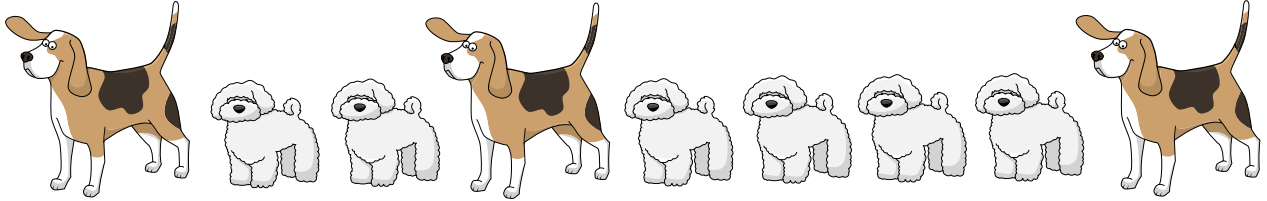
lettura dei grafi

- <https://it.wikipedia.org/wiki/Grafo>

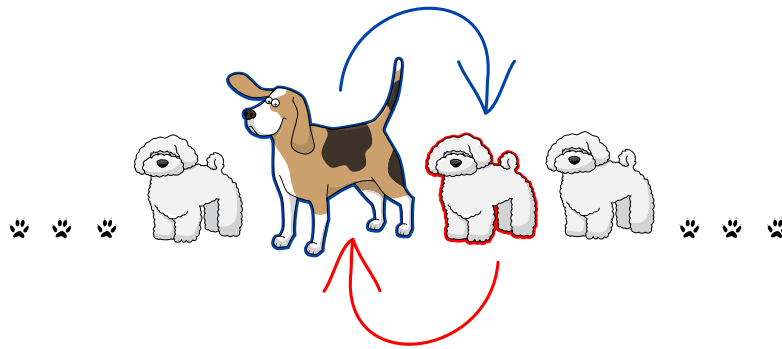


## 7. Scambio di cani

I cani di 2 razze diverse sono allineati su una fila:



Uno scambio consiste nell'invertire la posizione di 2 cani vicini.



Dopo una serie opportuna di scambi, i tre cani di grossa taglia sono posizionati uno vicino all'altro.  
*Qual è il numero minore di scambi necessario?*

- A) 5
- B) 6
- C) 7
- D) 8



## Soluzione

La risposta corretta è B) (6). Dapprima bisogna spostare il primo cane di grossa taglia 2 volte verso destra (2 scambi) e quindi l'ultimo cane di grossa taglia 4 volte verso sinistra (4 scambi).

Ogni cane di piccola taglia viene scambiato una volta, visto che ognuno di essi si trova tra due cani di grossa taglia. Uno scambio tra due cani di piccola taglia non ha nessun effetto, dunque bisogna considerare solo gli scambi tra una cane di grossa taglia e uno di piccola taglia. Dato che in totale ci sono 6 cani di piccola taglia, il numero minimo di scambi deve essere 6.

Se si volessero poi mettere i cani di grossa taglia tutti all'inizio o tutti alla fine della fila, allora sarebbero necessari almeno altri 6 scambi.

## Questa è l'informatica!

In una situazione qualsiasi, i cani potrebbero essere lasciati liberi di spostarsi casualmente finché quelli di grossa taglia non si trovano vicini. Nel nostro compito, però, essi rappresentano dei dati presenti nella memoria di un computer.

Quanto il computer deve ordinare delle informazioni senza avere molto spazio a disposizione, di solito scambia i dati (cani) a due a due. In informatica, questo viene detto *swap* ("scambio").

Nel nostro caso, il computer deve essere il più veloce possibile e dunque deve limitare il numero totale di scambi tra i cani per ottenere l'ordinamento desiderato. Sebbene un computer possa eseguire uno scambio molto più velocemente di qualsiasi essere umano, dobbiamo pensare che esso lavora con una quantità enorme di dati e dunque è necessario annullare le perdite di tempo. Questo significa ad esempio che un algoritmo efficiente potrebbe ordinare dei dati in mezzo secondo anziché in 2 o più minuti. In problemi computazionali complessi, un algoritmo ben programmato significa la differenza tra "pochi giorni" di calcolo e "anni interi".

## Parole chiave e siti web

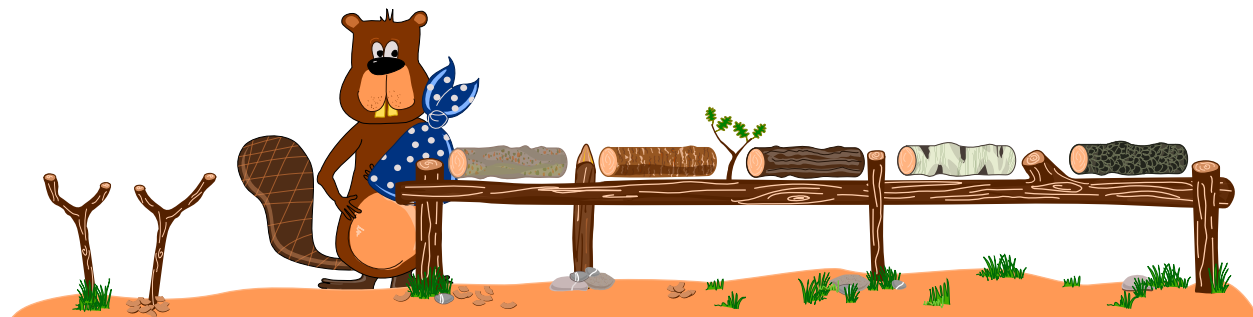
swap ("scambio")

- [https://en.wikipedia.org/wiki/Swap\\_\(computer\\_programming\)](https://en.wikipedia.org/wiki/Swap_(computer_programming))



## 8. Il castoro infortunato

David si è purtroppo rotto un braccio e al momento può utilizzare solo l'altro. Egli desidera mettere in ordine la sua collezione di ciocchi di legno, ma ne può sollevare solo uno alla volta. David può comunque spostare un ciocco nel sostegno a sinistra.

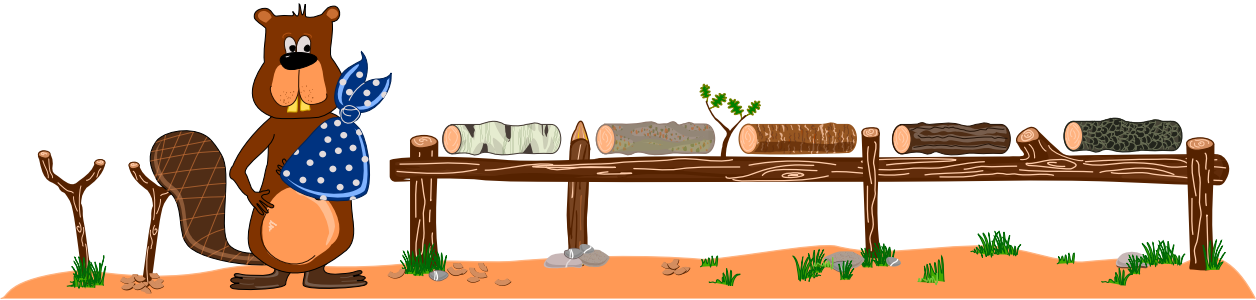


*Aiuta David ad ordinare la propria collezione di ciocchi di legno dal più chiaro al più scuro, in modo che alla fine quello più chiaro sia a sinistra e quello più scuro a destra.*



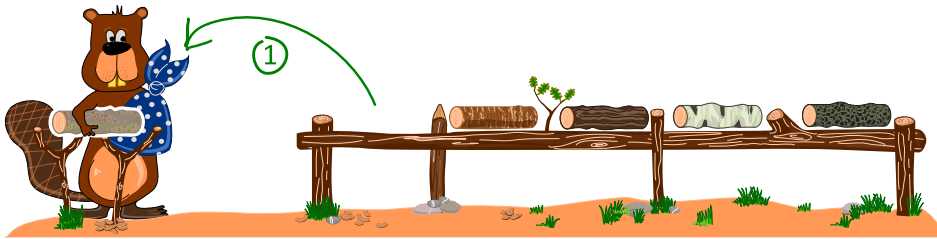
## Soluzione

La risposta corretta è:

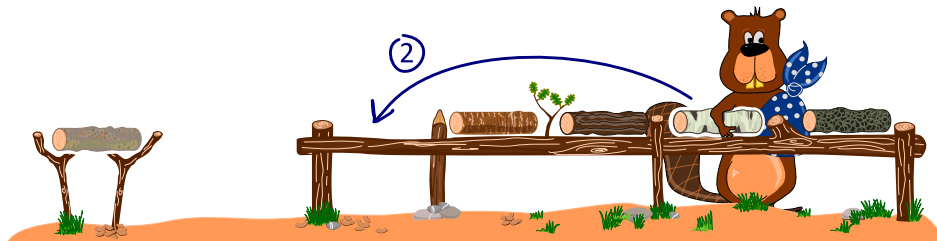


Per poter risolvere questo compito, David deve necessariamente utilizzare il sostegno a sinistra. Esistono molti metodi per ordinare i vari ciocchi, ma in ogni caso dobbiamo sempre scambiare di posto due ciocchi alla volta utilizzando il sostegno:

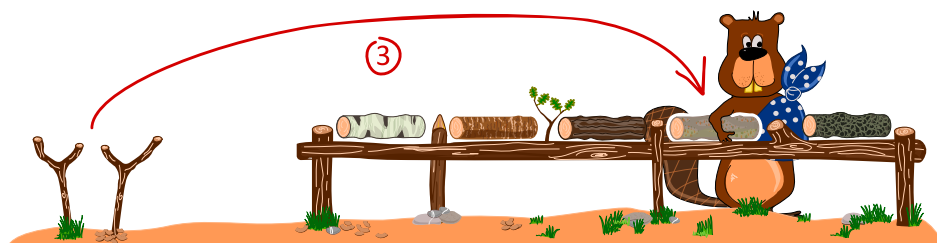
- Dapprima spostiamo un ciocco dalla sua posizione al sostegno.



- Poi un altro ciocco nella posizione lasciata libera.



- Quindi il primo ciocco dal sostegno alla posizione lasciata libera adesso.



In questo modo possiamo eseguire tutti gli scambi necessari a raggiungere l'ordine voluto.





## Questa è l'informatica!

Il principio per cui due ciocchi di legno vengono scambiati di posto, utilizzando un terzo spazio a disposizione è molto utilizzato in informatica. Gli spazi sono solitamente delle “variabili” (ad esempio  $a$  e  $b$ ) che contengono dei valori (“i ciocchi”). Se si vuole scambiare questi valori, si necessita di una terza variabile  $t$ , nella quale salvare temporaneamente il valore di  $a$ . Quindi possiamo assegnare ad  $a$  il valore di  $b$  e infine a  $b$  il valore contenuto in  $t$  (quello originale di  $a$ ). Possiamo descrivere questo procedimento anche con le espressioni:

$$\begin{aligned}t &\leftarrow a \\a &\leftarrow b \\b &\leftarrow t\end{aligned}$$

Per ordinare un insieme (*array*) composto da più variabili, disponendo di un solo spazio temporaneo, si può utilizzare ad esempi l'algoritmo di ordinamento per selezione (*selection sort*).

- Dalla prima all'ultima variabile:
  - Scegli il valore più piccolo tra le variabili non ancora scelte
  - Scambia questo valore con la variabile considerata, utilizzando la variabile temporanea

L'ordinamento per selezione non è l'unico algoritmo semplice esistente. Spesso si usa anche l'ordinamento per inserimento (*insertion sort*). Un algoritmo più efficiente, ma molto complicato, è il *quick sort* (ordinamento veloce).

## Parole chiave e siti web

Scambio di variabili, (algoritmi di) ordinamento

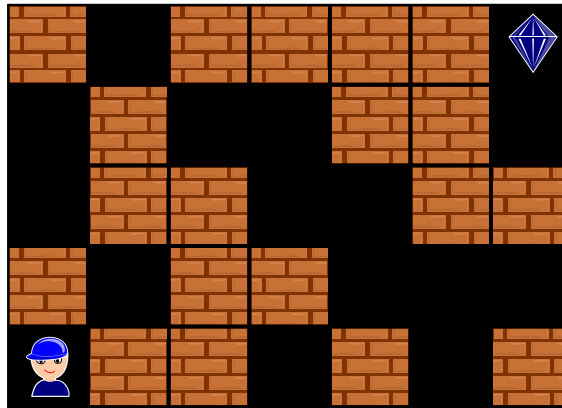
- [https://en.wikipedia.org/wiki/Swap\\_\(computer\\_programming\)](https://en.wikipedia.org/wiki/Swap_(computer_programming))
- [https://it.wikipedia.org/wiki/Algoritmo\\_di\\_ordinamento](https://it.wikipedia.org/wiki/Algoritmo_di_ordinamento)





## 9. Abbattere muri

Per raggiungere il diamante in alto a destra, Peter deve abbattere dei muri. Egli desidera però abbatte il meno possibile.

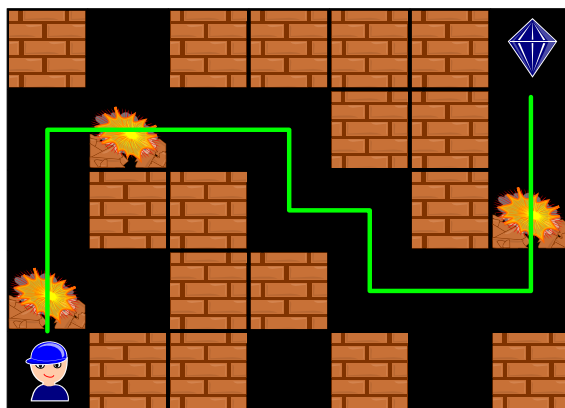


*Elimina il minor numero possibile di muri, per liberare la strada fino al diamante.*

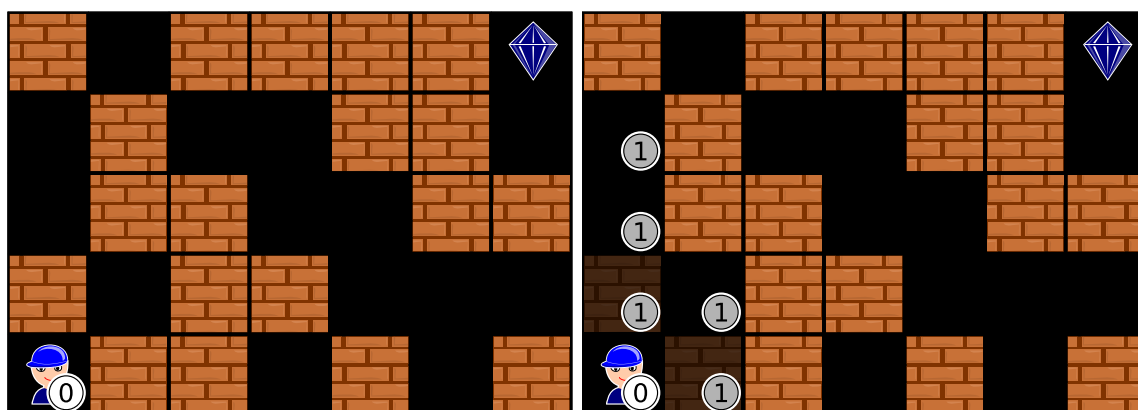


## Soluzione

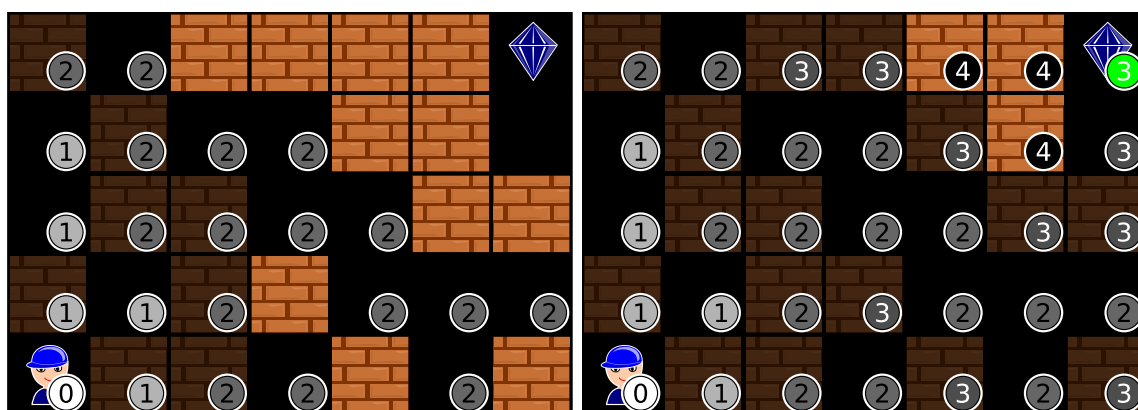
Peter deve abbattere almeno 3 muri. Nella figura qui sotto vediamo quali muri dove abbattere per liberare la strada fino al diamante.



Si può arrivare alla soluzione in modo sistematico, indicando in ogni cella il numero minimo di muri che è necessario abbattere per arrivarci. Iniziamo dalla posizione di Peter e segniamo 0 in questa cella. I due muri che lo circondano sono segnati con 1, così come tutte le celle nere che possono essere raggiunte quando essi vengono abbattuti.



Segniamo ora con 2 tutti i muri raggiungibili e, successivamente, tutte le celle nere percorribili dopo la loro demolizione. Continuiamo a procedere in questo modo fino a quando non viene liberata la strada per il diamante. Come vediamo la soluzione è 3.





## Questa è l'informatica!

La ricerca del *cammino minimo* (la strada più corta) in un labirinto è un problema informatico molto conosciuto. Anche per i sistemi di navigazione avvalgono di tecniche informatiche, per individuare il cammino minimo tra due punti, date certe condizioni (ad esempio, evitare le strade a pedaggio). Nel nostro compito non è la lunghezza del tragitto a dover essere minimizzata, ma il numero di muri da abbattere. Il problema è però lo stesso.

La soluzione presentata richiede un pensiero logico e l'elaborazione di un procedimento (algoritmo) nel quale si considerano, passo dopo passo, tutte i risultati parziali, fino a giungere alla soluzione. Questo procedimento è detto *ricerca in ampiezza*.

## Parole chiave e siti web

cammino minimo, labirinto, ricerca in ampiezza

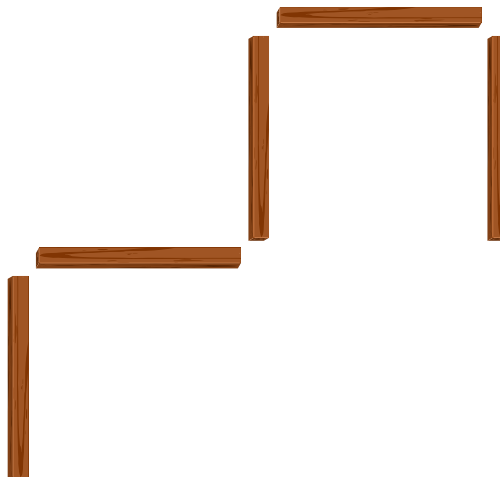
- [https://it.wikipedia.org/wiki/Cammino\\_minimo](https://it.wikipedia.org/wiki/Cammino_minimo)
- [https://it.wikipedia.org/wiki/Ricerca\\_in\\_ampiezza](https://it.wikipedia.org/wiki/Ricerca_in_ampiezza)



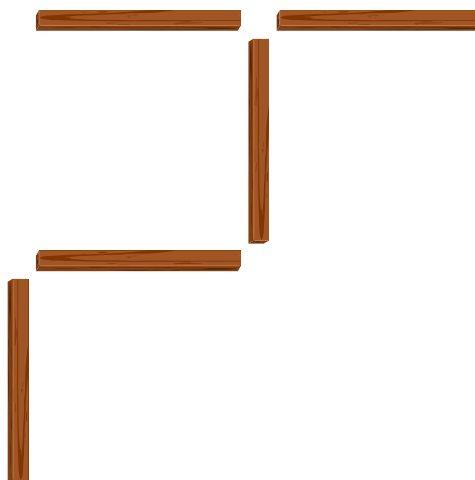


## 10. 5 bastoncini

5 bastoncini sono posti su un tavolo:

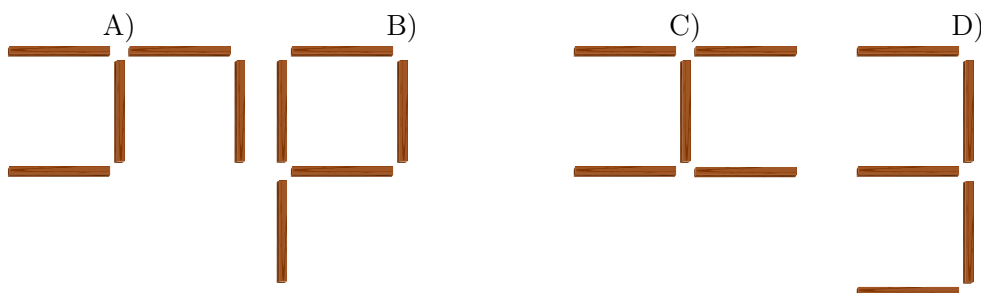


Nicole ne prende uno e lo mette in una nuova posizione. Adesso i bastoncini sono quindi collocati in questo modo:



Dopo Nicole, Roberto esegue un'operazione simile, prendono un altro bastoncino e posandolo in un posto diverso.

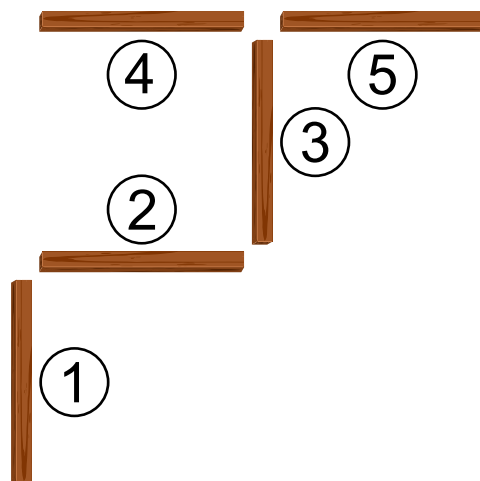
*In quale delle seguenti posizioni non è possibile che siano collocati i bastoncini, adesso?*





## Soluzione

Per spiegare la soluzione, numeriamo i bastoncini. Dopo l'operazione di Nicole, i bastoncini sono collocati in questo modo:



Per ottenere la collocazione di A), Roberto deve spostare il bastoncino 1.

Per ottenere la collocazione di B), Roberto deve spostare il bastoncino 5.

Per ottenere la collocazione di C), Roberto deve spostare il bastoncino 1.

Per ottenere la collocazione di D), Roberto dovrebbe spostare due bastoncini, ovvero il numero 1 e il numero 5. Ma Roberto ne ha spostato solo uno. D) è quindi la risposta!

## Questa è l'informatica!

Nicole e Roberto possono eseguire un'unica semplice operazione. Più precisamente, possono prendere e ricollocare un solo bastoncino per volta. Se volessimo impartire loro un ordine, diremmo dunque: "Sposta un bastoncino qualsiasi in un'altra posizione!". Questa istruzione non è però univoca, non viene infatti specificato quale bastoncino debbano prendere e quindi non sappiamo con sicurezza quale sarà la collocazione finale.

Quando si programmano dei computer, invece, bisogna sempre fornire delle istruzioni precise, comprese in modo inequivocabile dalla macchina. Nuove istruzioni possono essere create unendo una sequenza di istruzioni semplici.

## Parole chiave e siti web

istruzioni, cambiamento di stato, programmazione

- [https://it.wikipedia.org/wiki/Istruzione\\_\(informatica\)](https://it.wikipedia.org/wiki/Istruzione_(informatica))
- [https://it.wikipedia.org/wiki/Programmazione\\_\(informatica\)](https://it.wikipedia.org/wiki/Programmazione_(informatica))
- [https://it.wikipedia.org/wiki/Diagramma\\_di\\_stato\\_\(informatica\)](https://it.wikipedia.org/wiki/Diagramma_di_stato_(informatica))



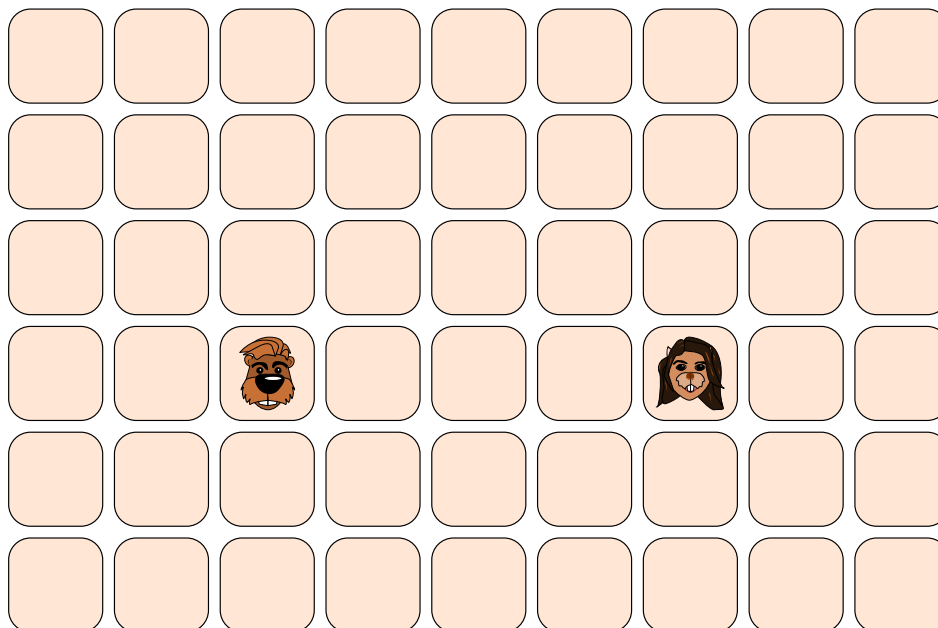


# 11. Esibizione di ballo

Uno scoiattolo (🐿️) e un castoro (🦫) partecipano ad un'esibizione di ballo. A seconda della reazione del pubblico, eseguono determinati passi. La tabella seguente mostra come si muovono:

	Wow!	Grida!	Applausi!	Fischi!
🐿️	← ↑	↑ ←	← ← ↑	↓ ↓
🦫	↑ →	→ ↓	↑ ↑ ↑	← ←

Per esempio, se il pubblico Grida!, lo scoiattolo si muove di una posizione verso l'alto e poi una verso sinistra; allo stesso tempo, il castoro si muove di una posizione verso destra e poi una verso il basso. I due partecipanti iniziano l'esibizione nelle seguenti posizioni:



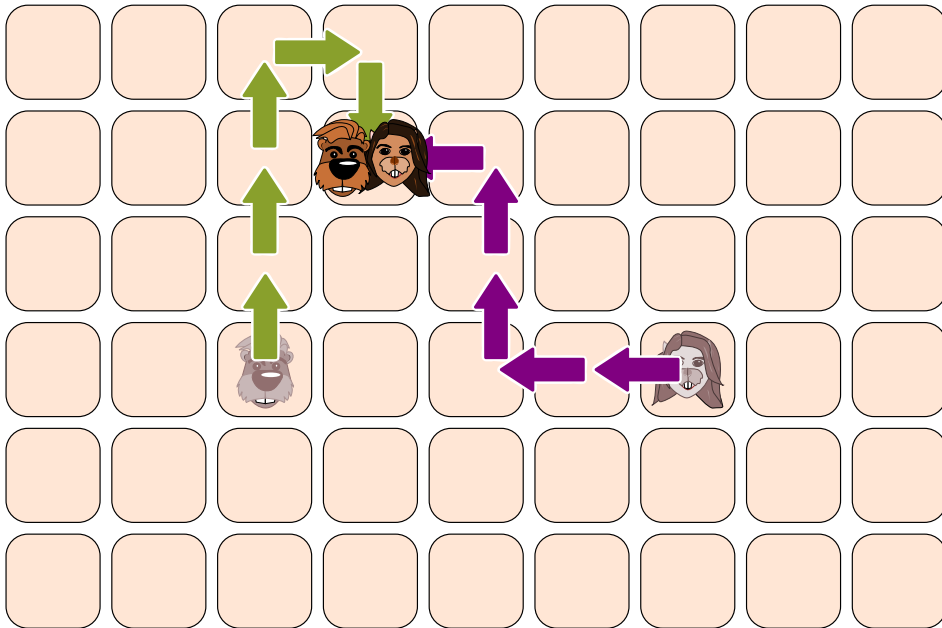
Quali delle seguenti reazioni del pubblico faranno in modo che lo scoiattolo e il castoro finiscano l'esibizione tutti e due nella stessa casella?

- A) Fischi! Grida!
- B) Wow! Grida!
- C) Grida! Grida!
- D) Applausi! Grida!



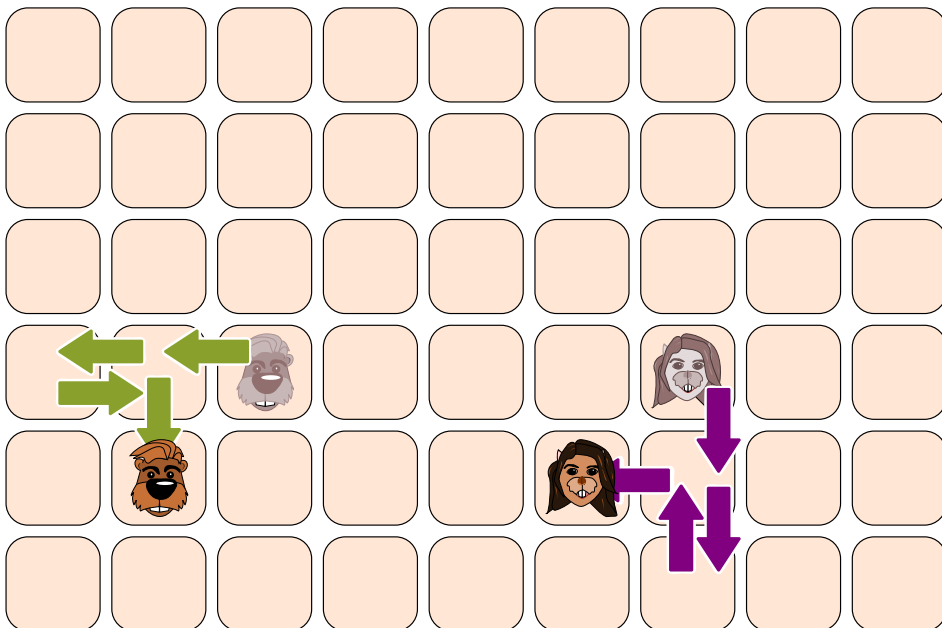
## Soluzione

La risposta D) “Applausi! Grida!” è corretta:



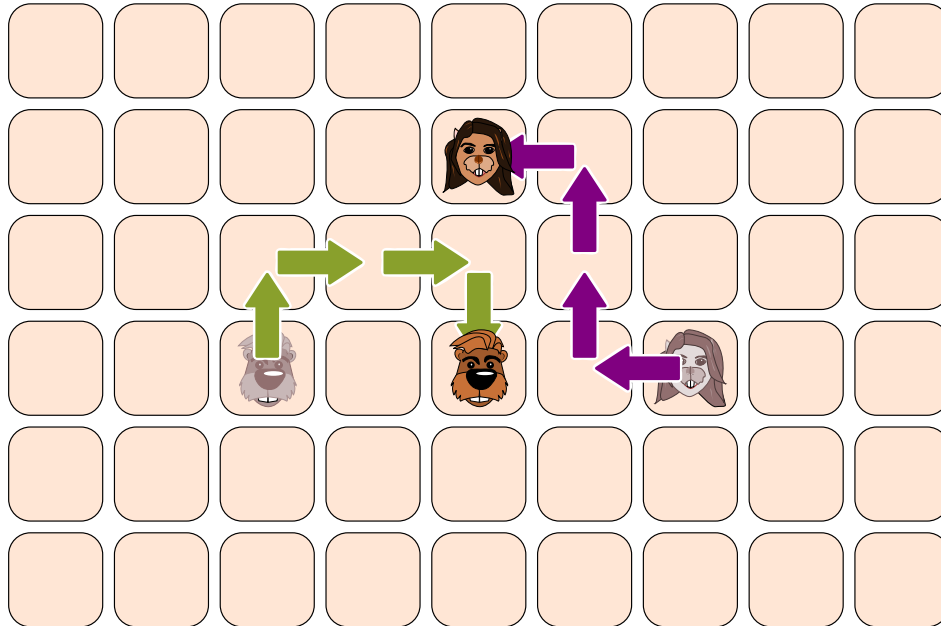
In tutti gli altri casi, lo scoiattolo e il castoro non terminano nella stessa casella:

A) “Fischi! Grida!”:

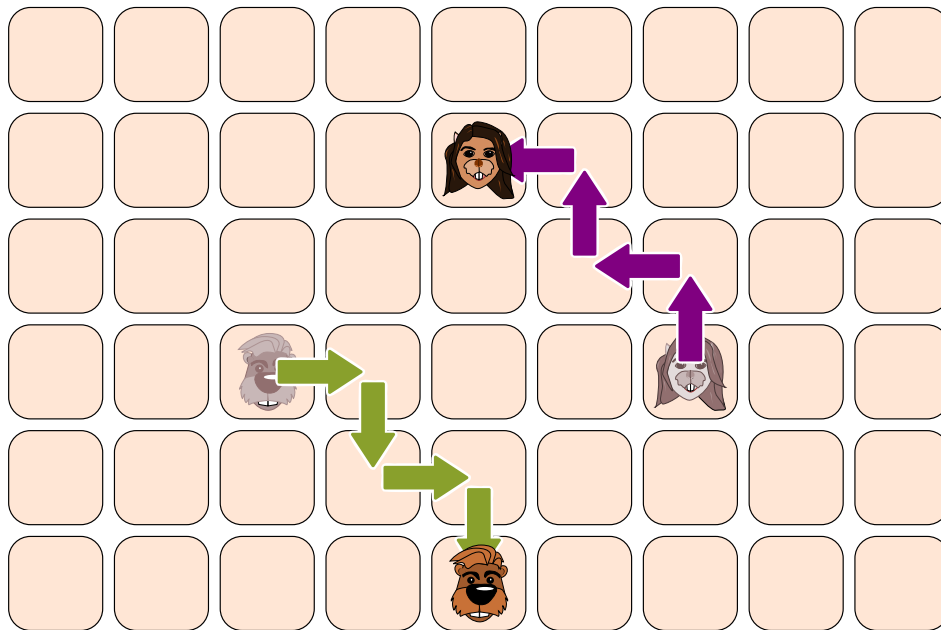




B) “Wow! Grida!”:



C) “Grida! Grida!”:



### Questa è l'informatica!

Questo compito è un esempio di *calcolo parallelo*. Nel nostro caso, i due “attori” si muovono indipendentemente uno dall'altro, osservando delle regole predefinite. Quando però i due attori collidono, ovvero quando vogliono accedere alla stessa risorsa contemporaneamente (come una posizione di memoria o una periferica collegata), bisogna decidere chi potrà disporne per primo, attraverso determinate regole. Ad esempio si potrebbero usare dei *semafori*: in questo modo un attore potrebbe prenotare, utilizzare e quindi liberare una determinata risorsa. Naturalmente, durante una prenotazione, nessun altro attore può accedere alla risorsa. Dunque, per evitare che essa possa essere prenotata da più parti contemporaneamente, esiste di norma un sistema centrale (il sistema operativo) che ha la responsabilità di gestire i semafori e le prenotazioni attraverso delle liste di attesa.



Mentre risolvevi l'esercizio, hai probabilmente eseguito delle *simulazioni* dei movimenti dei due attori. Le simulazioni eseguite al computer aiutano a capire meglio il mondo reale: ad esempio, è possibile fare delle previsioni del tempo accurate.

## Parole chiave e siti web

calcolo parallelo, semaforo, simulazioni

- [https://it.wikipedia.org/wiki/Calcolo\\_parallelo](https://it.wikipedia.org/wiki/Calcolo_parallelo)
- [https://it.wikipedia.org/wiki/Semaforo\\_\(informatica\)](https://it.wikipedia.org/wiki/Semaforo_(informatica))
- <https://it.wikipedia.org/wiki/Simulazione>



## 12. Il nome giapponese

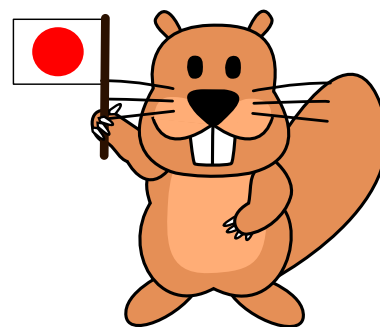
Un'amica giapponese ci racconta come, secondo una loro antica tradizione, dalle lettere del proprio nome si possa ottenere un nome spirituale, semplicemente sostituendo ogni lettera con delle sillabe associate:

A → ka	F → lu	K → me	P → mor	U → do	Z → zi
B → pi	G → ji	L → ta	Q → ke	V → ru	
C → mi	H → ri	M → rin	R → shi	W → mei	
D → te	I → ki	N → to	S → ari	X → na	
E → ku	J → zu	O → mo	T → chi	Y → fu	

Ad esempio, un tuo amico della Croazia ha ottenuto il seguente nome spirituale: "Zukame Moru".

*Quale è il vero nome del tuo amico?*

- A) Josip
- B) Jani
- C) Jakov
- D) Jurica





## Soluzione

Il tuo amico si chiama Jakov. Possiamo iniziare dalla prima sillaba, “Zu”, e quindi identificare la lettera “J”. La sillaba “ka” è “A”, ciò che già esclude Josip e Jurica dalle possibili risposte. La terza sillaba, “me”, corrisponde a “K” e dunque la risposta è necessariamente Jakov. Per essere sicuri, possiamo comunque facilmente verificare come “Moru” corrisponda ad “O” e “V”, cercando la traduzione di “O” (“mo”) e “V” (“ru”). Questa verifica avviene velocemente poiché le lettere sono ordinate alfabeticamente e dunque non dobbiamo cercare in tutta la tabella.

Potremmo trovare la soluzione ancora più velocemente se partissimo dal fondo: “ru” corrisponde infatti a “V” e l’unico nome che finisce con questa lettera è Jakov.

## Questa è l’informatica!

Probabilmente hai già sentito parlare di giochi simili. Spesso vengono utilizzati anche per creare dei codici segreti. L’idea alla base consiste nel sostituire in modo univoco ogni lettera di una parola con altre lettere o sillabe. In informatica questo procedimento viene chiamato *sistema di riscrittura di stringhe* (o parole) oppure *semi-sistema di Thue* (“*semi-Thue system*”).

Questi sistemi non costituiscono una base sicura per codici segreti. Essi fanno parte degli *algoritmi di cifratura monoalfabetici* e sono molto semplici da forzare, anche senza un computer.

Nel nostro compito, nella parola “Moru” possiamo identificare anche la sillaba “mor” che corrisponde a “P”. La sillaba “mo”, invece, corrisponde a “O” e costituisce un cosiddetto *prefisso* di “mor”. Se dovessimo creare un programma per la decodifica del nome spirituale, dovremmo stare molto attenti a come utilizziamo tali prefissi.

Tra parentesi, le sillabe utilizzate nel nostro compito sono realmente utilizzate in Giappone, per questo i nomi suonano così “orientali”.

## Parole chiave e siti web

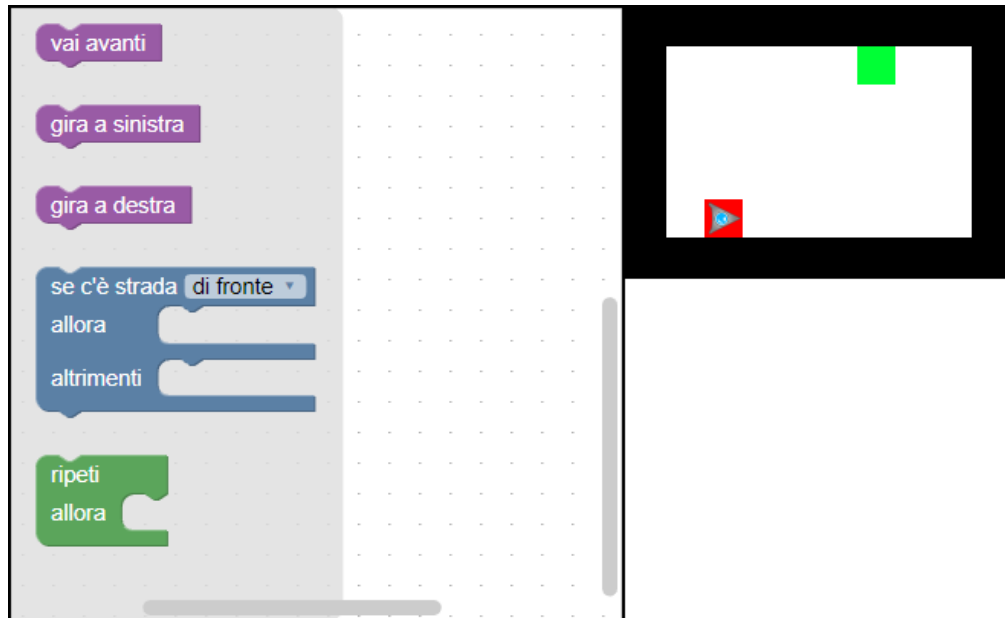
sistemi di riscrittura, semi-sistema di Thue, cifrario monoalfabetico

- [https://en.wikipedia.org/wiki/Semi-Thue\\_system](https://en.wikipedia.org/wiki/Semi-Thue_system)



## 13. Un piccolo programma

Un robot a forma triangolare si trova inizialmente nella posizione indicata in rosso e deve raggiungere il traguardo indicato in verde. Per fare questo può memorizzare solo un piccolo programma.

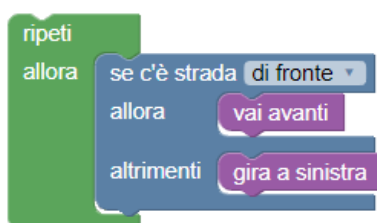


*Trascina le istruzioni nel pannello di programmazione e uniscile nella giusta sequenza per permettere al robot di raggiungere il traguardo. Puoi usare solo un massimo di 4 istruzioni!*



## Soluzione

Una possibile soluzione è:



L'idea base consiste nel far procedere il robot lungo il bordo (prima inferiore e poi quello a destra) e farlo girare ad ogni angolo verso sinistra. In questo modo arriverà in modo quasi “automatico” a destinazione.

## Questa è l'informatica!

Nel campo formato da  $8 \times 5$  celle, il robot ha bisogno di almeno 9 movimenti per raggiungere il traguardo: avanti – avanti – avanti – avanti – a sinistra – avanti – avanti – avanti – avanti.

Il programma può però contenere solo 4 istruzioni e quindi dobbiamo necessariamente utilizzare un ciclo (loop), attraverso il quale è possibile dire al robot di ripetere una determinata sequenza di istruzioni, senza bisogno di inserirla nuovamente nel codice del programma. Inoltre, per poter controllare la svolta a sinistra del robot e non farlo collidere con la parete, dobbiamo anche usufruire di un'istruzione condizionale.

Il programma così scritto, non permette al robot di prendere la strada più corta. Inoltre, tale programma, non funziona in tutte le situazioni: se il traguardo non fosse lungo una parete, il robot non potrebbe mai arrivarci, date le condizioni.

Sequenze di istruzioni, cicli, e istruzioni condizionali costituiscono la base di ogni algoritmo.

## Parole chiave e siti web

programmazione, cicli (loop), istruzioni condizionali, istruzioni

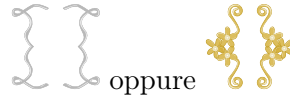
- <https://en.wikipedia.org/wiki/Blockly>
- <https://scratch.mit.edu/>
- <http://code.org/>
- <http://insegnamento.adamoli.ch/>



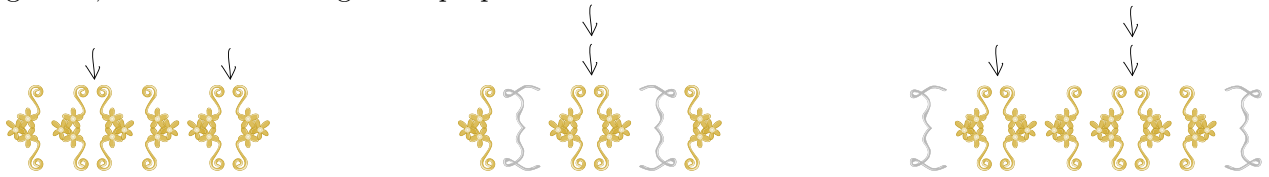


# 14. Gioielli

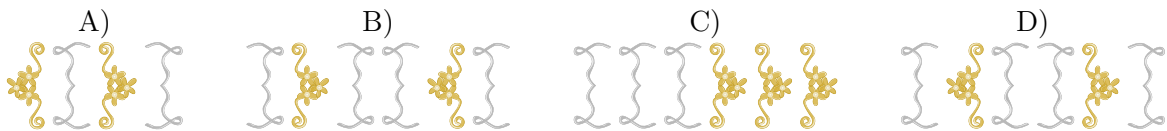
La famiglia Castoro produce artigianalmente dei gioielli in stile medioevale. Per fare questo, utilizza come base degli ornamenti a forma di parentesi, impiegati sempre a coppie. Per creare un gioiello, inizia da una delle seguenti coppie “di parentesi”:



Dopodiché, ripete più volte l’inserimento di una coppia “di parentesi” in un punto qualsiasi del gioiello, come mostrato negli esempi qui sotto:



Quali dei seguenti gioielli è stato creato con il metodo spiegato sopra?





## Soluzione

La risposta corretta è D). Nella coppia iniziale sono stati inseriti, uno dopo l'altro, due nuove coppie di ornamenti a forma di parentesi, sempre al centro.

Tutti gli altri gioielli non sono stati creati con il metodo spiegato:

- A) Da sinistra, l'errore sta in posizione 3: una parentesi viene chiusa, prima che quella in posizione 2 lo sia.
- B) Da sinistra, l'errore si trova già in posizione 1: la parentesi viene chiusa, prima di essere aperta.
- C) Da sinistra, l'errore si trova in posizione 4: una parentesi viene chiusa, senza mai essere stata aperta.

## Questa è l'informatica!

Le regole per la produzione dei gioielli sono esattamente le stesse che si usano per le parentesi in matematica o in informatica. Espressioni senza errore vengono dette "*ben formate*" ("*well-formed*"). Un'espressione ben formata è anche detta "*sintatticamente corretta*", se soddisfa le regole sintattiche della grammatica di una qualsiasi lingua formale (ad esempio, un linguaggio di programmazione). Errori di sintassi ("*syntax errors*") sono in genere più semplici da correggere, rispetto ad altri, talvolta molto subdoli, generati da errori di pensiero. Quest'ultimi sono chiamati "*errori semantici*".

## Parole chiave e siti web

espressione "ben formata", sintassi, semantica

- <https://it.wikipedia.org/wiki/Sintassi>
- <https://it.wikipedia.org/wiki/Semantica>



## 15. Il giornale della scuola

Al giornale della scuola lavorano 10 volontari. Ogni Venerdì utilizzano le loro “ore buche” per scrivere gli articoli. L’orario scolastico mostra in verde quando i diversi volontari hanno tempo per il giornale.

	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00
Anna								
Bea								
Celine								
David								
Emma								
Flo								
Gioa								
Hans								
Ida								
Jakob								

Per il lavoro hanno bisogno di nuovi notebook che devono essere acquistati dalla scuola.

*Quanti notebook nuovi deve acquistare la scuola, in modo da permettere ad ognuno dei volontari di lavorarci durante il proprio turno?*

- A) quattro
- B) cinque
- C) sette
- D) dieci



## Soluzione

La risposta A) è sbagliata. Tra le 9:00 e le 10:00, ad esempio, devono poter lavorare 5 persone (1, 3, 5, 8 e 9) contemporaneamente e quindi non ci sono abbastanza notebook.

La risposta B) è giusta. Dalle 9 alle 10 e dalle 10 alle 11 lavorano esattamente 5 volontari. Questo è il numero massimo di persone che lavorano allo stesso momento, in tutte le altre ore ci sono meno volontari. Dunque 5 è il numero massimo di notebook necessario.

	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00
Anna		3	3					
Bea			1	1	1	1		
Celine	1	1						
David					3	3	3	
Emma		2	2					
Flo				2	2			
Gioa			5	5		5	5	
Hans		5						
Ida	4	4	4					
Jakob						2	2	

Dato che cinque notebook sono sufficienti, le risposte C) e D) sono sbagliate.

## Questa è l'informatica!

Quando bisogna comprendere le inter-dipendenze di un numero enorme di dati è molto importante scegliere la giusta rappresentazione come ad esempio una tabella, un diagramma o un grafo. A dipendenza dello scopo, le diverse rappresentazioni sono più o meno vantaggiose.

Per il nostro compito, vale la pena rappresentare i dati in una tabella come segue:

	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00
1	Celine	Celine	Bea	Bea	Bea	Bea		
2		Emma	Emma	Flo	Flo	Jakob	Jakob	
3		Anna	Anna		David	David	David	
4	Ida	Ida	Ida					
5		Hans	Gioa	Gioa		Gioa	Gioa	

In questa rappresentazione, i notebook vengono associati, per ogni ora, a una persona. In questo modo possiamo verificare velocemente che 5 notebook sono sufficienti.

Simili pianificazioni sono spesso impiegate quando si tratta di utilizzare delle risorse in modo ottimale. Ogni hotel, ad esempio, possiede un sistema analogo per la prenotazione delle stanze: logicamente le prenotazioni non possono sovrapporsi e bisogna fare in modo di sfruttare tutte le stanze al meglio. Lo stesso discorso vale per le aule scolastiche, anche se "ottimale", in questo caso, potrebbe avere significati diversi: ad esempio, è meglio fare in modo che una classe stia sempre nella stessa aula, oppure che il docente debba cambiare aula il meno possibile?

Un ulteriore tipo di rappresentazione per tali pianificazioni è il grafo d'intervallo, in cui i nodi sono gli intervalli di tempo e gli archi le loro sovrapposizioni.

## Parole chiave e siti web

pianificazione, grafo d'intervallo



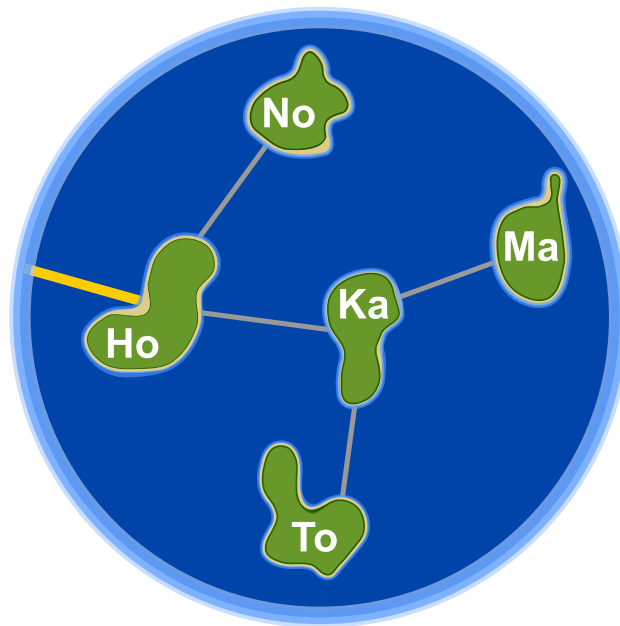
- [https://it.wikipedia.org/wiki/Grafo\\_d'intervallo](https://it.wikipedia.org/wiki/Grafo_d'intervallo)





## 16. Honomakato

L'arcipelago Honomakato è formato dalle cinque isole Ho, No, Ma, Ka e To. L'isola principale Ho è collegata a Internet tramite un cavo molto affidabile. Inoltre tra Ho e No, Ho e Ka, Ka e Ma, Ka e To sono collocati ulteriori cavi. Tutte le isole sono dunque collegate con Ho e quindi anche con Internet.



Gli abitanti di Honomakato desiderano avere una connessione affidabile a Internet per tutte le isole: in altre parole, anche se un cavo tra di esse dovesse danneggiarsi, ogni isola minore dovrebbe sempre poter accedere a Internet.

*Fai in modo che ogni isola abbia un collegamento affidabile a Internet. Piazza 2 ulteriori cavi tra le isole.*

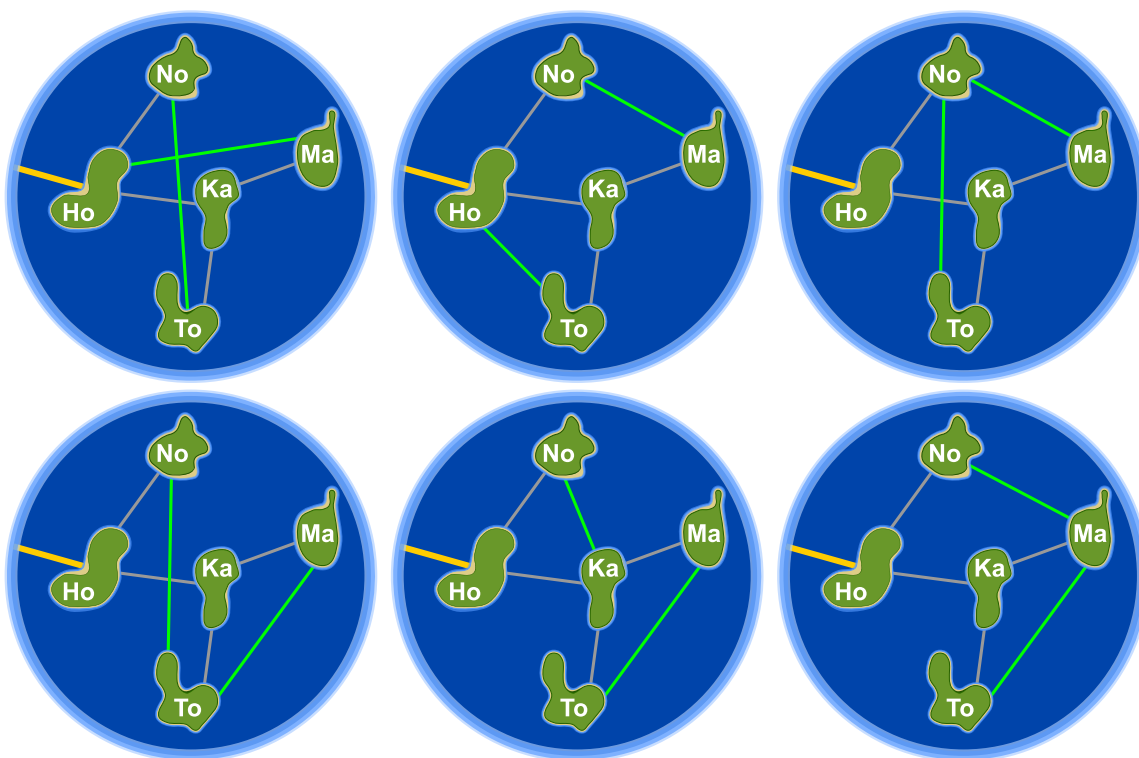


## Soluzione

Con la posa di 2 ulteriori cavi, ci si può assicurare che l'arcipelago di Honomakato possieda un collegamento a Internet affidabile. Esistono sei diverse possibilità per permettere alle isole minori di collegarsi a Internet anche se un cavo dovesse danneggiarsi.

1. Ho-Ma e No-To: Ho-Ma protegge Ma e Ka, No-To protegge No e To dalle rotture.
2. Ho-To e No-Ma: Ho-To protegge To e Ka, No-Ma protegge Ma, No, e Ka dalle rotture.
3. No-To e No-Ma: No-To protegge No, To e Ka, No-Ma protegge Ma, No, e Ka dalle rotture.
4. No-To e Ma-To: No-To protegge No, To e Ka, Ma-To protegge Ma e To dalle rotture.
5. No-Ka e Ma-To: No-Ka protegge No e Ka, Ma-To protegge Ma e To dalle rotture.
6. No-Ma e Ma-To: No-Ma protegge Ma, No, e Ka, Ma-To protegge Ma e To dalle rotture.

In generale, per ogni soluzione sono soddisfatte le regole seguenti: (1) per ogni isola esistono almeno 2 connessioni e (2) l'arcipelago di Honomakato non può essere suddiviso in 2 gruppi qualsiasi connessi da un unico cavo.



## Questa è l'informatica!

Le rete di cavi con la quale l'arcipelago di Honomakato è collegato a Internet rappresenta una piccola parte della rete globale e un esempio di come essa è costruita. I router, i server e altri dispositivi telematici sono i nodi di una grossa rete chiamata Internet, esattamente come le isole lo sono nel nostro arcipelago.

Internet è stato concepito negli anni '60 come rete robusta (ossia, "affidabile"). Un guasto alle singole connessioni tra i nodi non deve in alcun modo pregiudicare il funzionamento dell'intera rete.





Pertanto ogni nodo dispone di connessioni multiple e in caso di rottura o congestione di una di esse, le altre possono essere utilizzate come riserva. Anche per altre reti (come quelle di trasporto o quelle di approvvigionamento) è importante che ogni nodo non venga isolato in caso di rottura di una connessione.

L'informatica utilizza la teoria dei grafi per eseguire calcoli su questo tipo di reti. Un grafo è definito da una rete di nodi e di collegamenti (detti archi) tra essi. Un grafo è detto “connesso” se per ogni coppia di nodi A e B, B è collegato ad A da un percorso attraverso uno o più archi. Un singolo arco necessario affinché un grafo possa essere detto connesso, viene chiamato “ponte”. In informatica sono stati sviluppati degli algoritmi per individuare questi ponti, in particolare Robert Tarjan ne ha sviluppato uno molto efficiente.

## Parole chiave e siti web

Strutture di dati dinamiche, grafo, bridge (“ponte”)

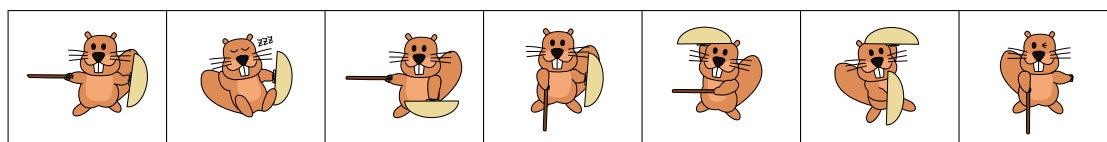
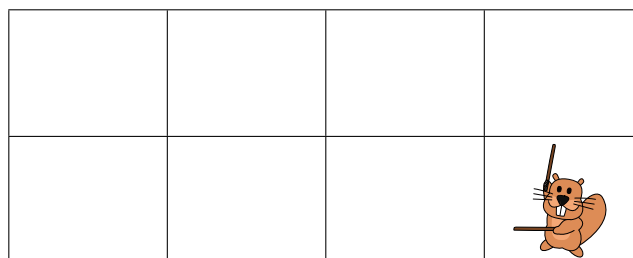
- [https://en.wikipedia.org/wiki/Vertex\\_separator](https://en.wikipedia.org/wiki/Vertex_separator)





## 17. L'arte del bastone giapponese

Lucia e i suoi amici fanno parte di un club che pratica l'arte del bastone giapponese. Per una foto, desiderano sedersi nel cortile scolastico in modo che ogni bastone punti verso uno scudo. Per aiutarsi a fare questo, hanno disegnato diversi campi nel cortile. Lucia si è già messa in posa e sotto a lei puoi vedere tutti i suoi amici che mostrano la propria posa preferita.

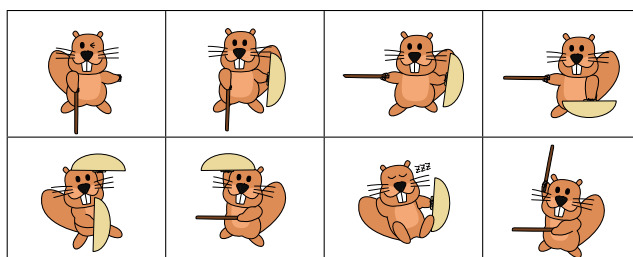


Associa le immagini degli amici di Lucia ai campi del cortile in modo che nella foto ogni bastone punti verso uno scudo.



## Soluzione

La risposta corretta è:



Le immagini degli amici di Lucia devono essere spostate nei campi vuoti esattamente come mostrato: così ogni bastone punta verso uno scudo. Non esistono altre possibilità.

## Questa è l'informatica!

In totale dobbiamo collocare 7 immagini nella giusta sequenza. Se proviamo a risolvere il compito in modo casuale, sprechiamo molto tempo. Esistono in fatti  $1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 = 7! = 5040$  diverse disposizioni e la maggior parte sono ovviamente errate. Con un po' di logica è però possibile trovare molto più velocemente la soluzione:

1. Tutti i castori che puntano un bastone o uno scudo verso l'alto devono necessariamente stare nella riga in basso.
2. Tutti i castori che puntano un bastone o uno scudo verso il basso, devono necessariamente stare nella riga in alto.
3. Esiste un solo castoro che punto il proprio scudo verso il basso, esso deve quindi essere posizionato sopra a Lucia. kann.

Con queste considerazioni, lo spazio di ricerca per la soluzione corretta si riduce a poche possibilità. Il procedimento per cui tutte le soluzioni possibili vengono provate secondo il principio "prova e sbaglia" si chiama *backtracking*. Tale procedimento è utilizzabile solo se lo spazio di ricerca è ragionevolmente ridotto. Fare le giuste considerazioni logiche è pertanto molto importante.

## Parole chiave e siti web

pensiero logico, conclusioni logiche

- <https://it.wikipedia.org/wiki/Backtracking>



## 18. La marmellata della nonna

Mentre la nonna prepara la marmellata, Anna, Peter e Lisa la aiutano a riempire i barattoli. Per fare questo devono compiere in sequenza ordinata queste operazioni:



Pulire il barattolo:  
durata 3 minuti.



Riempire il barattolo con la marmellata: durata 2 minuti.



Sigillare il barattolo: durata 1 minuto.

Anna, Peter e Lisa desiderano spartirsi il lavoro e quindi elaborano un piano. Nel fare questo devono state attenti a completare ogni singola operazione prima di iniziare quella successiva: un barattolo deve essere sempre pulito, prima di poterlo riempire e deve essere pieno prima di poterlo sigillare. Il piano seguente, non è dunque possibile:



ANNA										
PETER										
LISA										

Anna Peter e Lisa vogliono completare più barattoli possibili, in 10 minuti. Elabora il piano migliore possibile per questo lavoro:

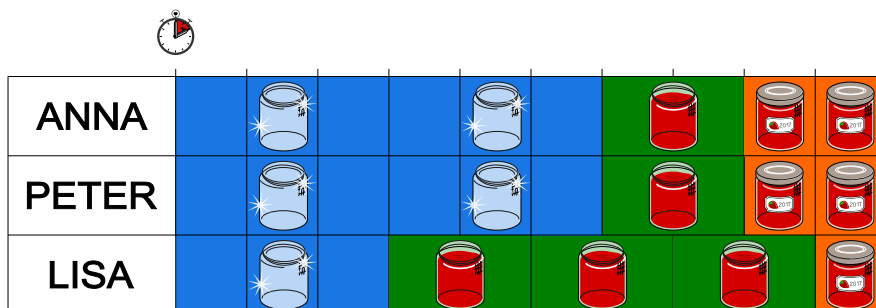


ANNA										
PETER										
LISA										



## Soluzione

Con il piano mostrato qui, Anna, Peter e Lisa riescono a completare 5 barattoli in 10 minuti. Esistono comunque altri piani che consentono a loro di ottenere lo stesso risultato, ad esempio scambiando di posto i 3 bambini.



Non si possono riempire più di 5 barattoli in 10 minuti. Il completamento di un barattolo richiede 6 minuti ( $3 + 2 + 1 = 6$ ) ed ogni bambino ha a disposizione 10 minuti di lavoro. In 30 minuti totali possiamo dunque riempire un massimo di  $30/6 = 5$  barattoli.

## Questa è l'informatica!

In questo compito Anna, Peter e Lisa lavorano allo stesso tempo, quindi eseguono in *parallelo* ogni singola operazione. L'esecuzione di lavori in parallelo è un tema molto importante in informatica. I centri di calcolo, come il CSCS di Lugano, posseggono molti calcolatori che lavorano in parallelo sullo stesso problema, suddividendo opportunamente il lavoro in tante piccole operazioni. Anche nei computer domestici e negli smartphone coesistono più processori, in grado di lavorare parallelamente. La pianificazione del lavoro in informatica è detta *scheduling*.

Anche in altre attività, non solo in informatica, ci sono problemi di scheduling. Per esempio la pianificazione di grossi progetti richiede spesso l'utilizzo di un diagramma delle precedenze. In esso viene determinata la giusta sequenza in cui le varie operazioni devono essere eseguite e quanto tempo bisogna riservare per ognuna di esse.

## Parole chiave e siti web

Scheduling (pianificazione), parallelismo (esecuzione parallela)

- [https://en.wikipedia.org/wiki/Precedence\\_diagram\\_method](https://en.wikipedia.org/wiki/Precedence_diagram_method)
- <https://it.wikipedia.org/wiki/Scheduler>
- <http://www.cscs.ch/>

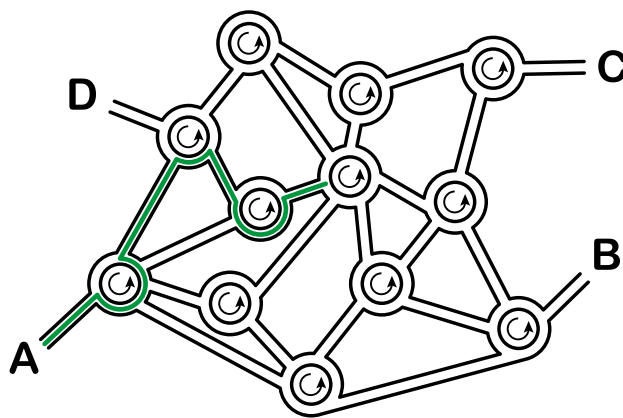


## 19. La città delle rotonde

Nella città dei castori non esistono incroci, ma solo rotonde. Quando gli abitanti devono spiegare la strada a qualcuno, dicono ad esempio:

- Alla prossima rotonda, prendi la 4<sup>a</sup> uscita.
- Alla rotonda successiva, prendi la 1<sup>a</sup> uscita.
- A quella dopo, prendi la 2<sup>a</sup> uscita.

Se la persona conosce già la città, allora i castori dicono solamente “4 1 2”, poiché è chiaro cosa intendono.



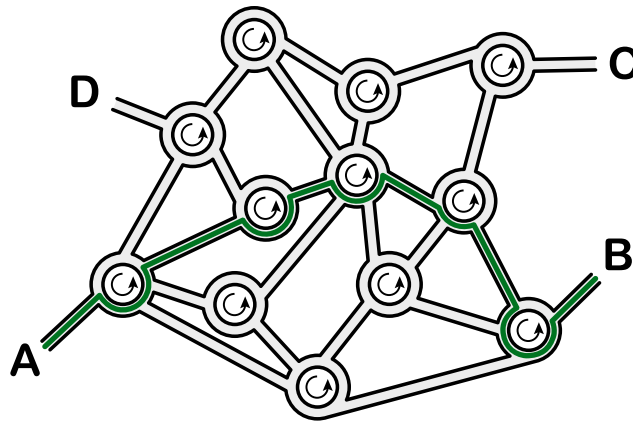
*Dove conduce la spiegazione “3 1 3 2 3”, partendo dal punto A?*

- A) Conduce al punto A.
- B) Conduce al punto B.
- C) Conduce al punto C.
- D) Conduce al punto D.



## Soluzione

La spiegazione conduce al punto B.



## Questa è l'informatica!

Questo compito è un buon esempio di “informazione strutturata”. Un sistema di guida computerizzato non potrebbe mai comprendere le indicazioni dette nella nostra lingua. Quando, però, strutturiamo queste indicazioni ad esempio in una sequenza di numeri, le informazioni assumono una forma che il computer può facilmente interpretare. Le sequenze di istruzioni (strutturate) sono alla base di molti linguaggi di programmazione.

Nel nostro compito è utile che le rete stradale sia molto simile: tutti gli incroci, infatti, sono formati da rotonde. Strutture come queste vengono dette omogenee, in antitesi rispetto a quelle dette eterogenee. Strutture omogenee sono molto più semplici da formalizzare che le strutture eterogenee, per questo sono preferite dagli informatici.

## Parole chiave e siti web

sequenze (di istruzioni), esecuzione di programmi, linguaggi formali

- [https://it.wikipedia.org/wiki/Programmazione\\_imperativa](https://it.wikipedia.org/wiki/Programmazione_imperativa)

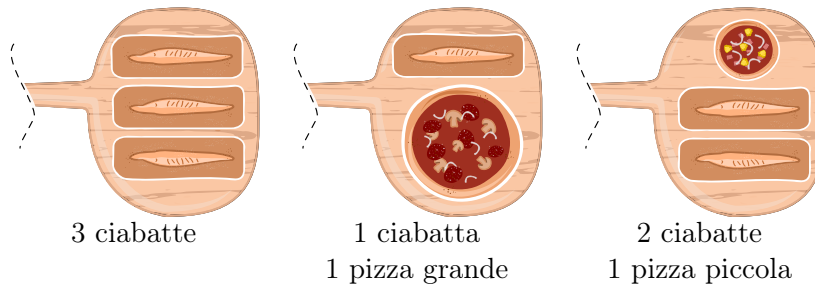




## 20. La pizzeria dei castori

La pizzeria dei castori possiede un solo forno a legna e quindi solo pochi cibi possono essere cotti contemporaneamente.

Le combinazioni sono:



I cibi necessitano di diversi tempi di cottura: una pizza piccola deve essere cotta per 10 minuti, una pizza grande per 15 minuti, mentre una ciabatta per 20 minuti. Il pizzaiolo può comunque mettere o togliere dal forno i diversi cibi al momento più opportuno.

Oggi c'è molto da fare. Il pizzaiolo deve preparare una pizza piccola, due pizze grandi e quattro ciabatte. I clienti sono molto affamati e desiderano essere serviti al più presto possibile.

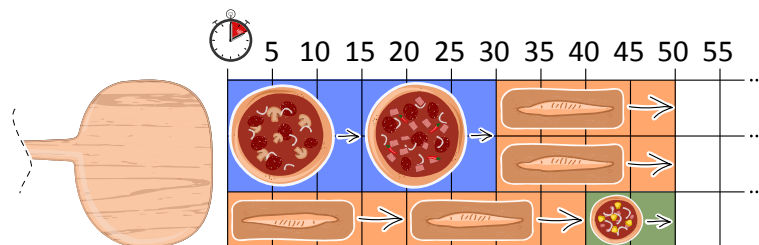
*Qual è il minor tempo in assoluto necessario a cuocere tutti i cibi?*





## Soluzione

Ci sono più soluzioni ottimali. Tra di esse, quella più semplice consiste nel cuocere le due pizze grandi una dopo l'altra in parallelo con due ciabatte. Infine si inseriscono nel forno le due ciabatte rimanenti e poi una pizza piccola:

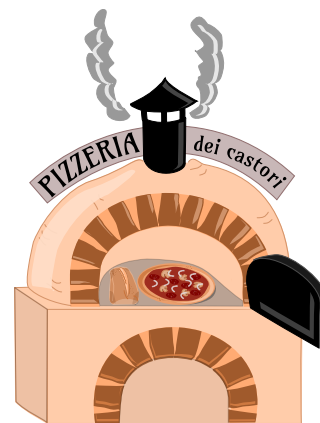


Queste soluzioni necessitano in totale di 50 minuti. Una soluzione più veloce non esiste, dato che il forno viene sempre utilizzato al massimo della sua capacità.

## Questa è l'informatica!

Quando si cerca di pianificare (in inglese *schedule*) l'utilizzo di risorse limitate (nel nostro caso il forno), l'obiettivo è in generale quello di trovare una soluzione ottimale in cui i costi o il tempo (come nel nostro caso) vengano minimizzati. In questo modo la risorsa viene sfruttata appieno. Un procedimento molto diffuso è quello dei turni (in inglese *round-robin*). In esso, i vari processi che devono essere eseguiti vengono inseriti in una lista di attesa. Ogni processo, a turno, riceve accesso alla risorsa per poco tempo. Se alla fine del proprio turno il processo non è ancora stato completato, esso viene inserito di nuovo alla fine della lista di attesa.

Per il nostro forno, questa strategia però non è utilizzabile. Non è possibile infatti togliere dal forno un cibo che non è stato completamente cotto per poi rimetterlo più tardi. Nel nostro caso è quindi meglio utilizzare un procedimento detto *greedy* ("avido"): si eseguono dapprima i processi che utilizzano più risorse (come la pizza grande, che usa due spazi nel forno) e di seguito quelli un po' più piccoli fino a completare il lavoro. Dato che il forno ha la capacità di cuocere più cibi per volta, è possibile ottimizzare lo spazio utilizzato togliendo i cibi cotti e inserendone dei nuovi al momento più opportuno.



Ma siamo sicuri di aver trovato una soluzione ottimale? Questo non è evidente, poiché non è facile escludere che esista una soluzione migliore a quella trovata. Quando però si riesce ad utilizzare le risorse al massimo delle proprie capacità in ogni momento (come nel nostro caso), possiamo affermare con sicurezza che non esiste alcuna soluzione migliore.

## Parole chiave e siti web

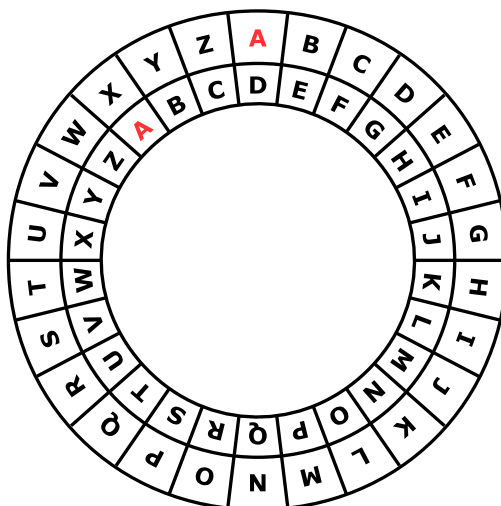
scheduling, liste di attesa, risorse, round-robin, algoritmo greedy

- <https://it.wikipedia.org/wiki/Scheduler>
- [https://en.wikipedia.org/wiki/Round-robin\\_scheduling](https://en.wikipedia.org/wiki/Round-robin_scheduling)
- [https://it.wikipedia.org/wiki/Algoritmo\\_greedy](https://it.wikipedia.org/wiki/Algoritmo_greedy)



## 21. Ordinazione segreta

Anna ordina al ristorante con messaggi segreti: solo Cesare, il cuoco, può decifrarli. Per scrivere i messaggi utilizza un disco particolare, composto da un anello esterno e uno interno mobile. Gli anelli sono a loro volta formati dalle lettere dell'alfabeto. All'inizio le lettere dei due anelli sono allineate: la "A" interna si trova sotto la "A" esterna, la "B" interna sotto la "B" esterna e così via.



Anna scrive il messaggio segreto in questo modo: dapprima scrive l'ordinazione, ad esempio PIZZA, poi esegue le seguenti operazioni:

1. Sotto ogni lettera dell'ordinazione, scrive un numero a caso che indica una rotazione.
2. Per ogni lettera del messaggio originale, Anna mette dapprima l'anello interno nella posizione iniziale e poi lo ruota in senso anti-orario di tante lettere quanto indicato dal numero di rotazione.
3. Nel messaggio segreto sostituisce ogni lettera originale con la lettera indicata dall'anello interno.

Per esempio, se Anna desidera ordinare la PIZZA e utilizza i numeri di rotazione 3, 1, 4, 1 e 5, invierà il messaggio segreto SJDAF.

Ordinazione	P	I	Z	Z	A
Numero di rotazione	3	1	4	1	5
Messaggio segreto	S	J	D	A	F

Per un'altra ordinazione, Anna ha scelto i numeri di rotazione 3, 1, 4, 1, 5, 9 e 2 e quindi ha inviato il messaggio segreto OBWBLWC.

*Cosa ha ordinato Anna?*

Ordinazione							
Numero di rotazione	3	1	4	1	5	9	2
Messaggio segreto	O	B	W	B	L	W	C



## Soluzione

La risposta corretta è LASAGNA:

Ordinazione	L	A	S	A	G	N	A
Numero di rotazione	3	1	4	1	5	9	2
Messaggio segreto	O	B	W	B	L	W	C

Possiamo risalire all'ordinazione utilizzando il nostro disco, ruotando questa volta l'anello interno in senso orario e indicando la lettera dell'anello esterno (ovvero facciamo il contrario di quanto fatto per codificare il messaggio).

## Questa è l'informatica!

Anna codifica il messaggio in modo che solo il suo cuoco preferito possa capire l'ordinazione. La crittografia è un metodo utilizzato fin dai tempi antichi. Il motivo è chiaro: vogliamo essere sicuri che il messaggio sia letto solo del destinatario designato e non dalle spie. Esistono molti metodi per la crittografia, ma in generale abbiamo sempre a che fare con due tipi di algoritmi principali: quello per la codifica e quello per la decodifica. Entrambi necessitano di una chiave per eseguire il lavoro. Un metodo per la crittografia tra i più semplici risale a Giulio Cesare: in esso, la chiave è un numero che indica uno spostamento all'interno dell'alfabeto. La chiave 3, ad esempio, indica che la lettera "A" viene codificata in una "D", "B" in "E", e così via. Allo stesso tempo, indica che la lettera "D" viene de-codificata in "A", "E" in "B", ecc. Il "disco di Cesare" (meglio, "cifrario di Cesare") mostrato in questo compito aiuta nel cifrare e decifrare i messaggi con questo procedimento.

Metodi per la cifratura che usano una chiave semplice (ad esempio di un solo numero) sono relativamente insicuri. Anna lo sa bene e per questo utilizza numeri diversi per ogni lettera, utilizzando dunque un metodo simile a quello di Vigenère. In questo procedimento i numeri che formano la chiave possono essere ripetuti quanto necessario per coprire la lunghezza del messaggio e dunque la chiave non è necessariamente troppo lunga. Ma anche questo metodo può risultare insicuro, soprattutto per chiavi corte e messaggi lunghi.



## Parole chiave e siti web

Crittografia, cifrario poli-alfabetico, cifrario di Cesare, cifrario di Vigenère

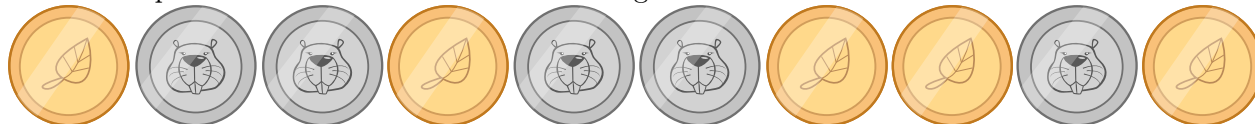
- [https://it.wikipedia.org/wiki/Cifrario\\_di\\_Cesare](https://it.wikipedia.org/wiki/Cifrario_di_Cesare)
- [https://it.wikipedia.org/wiki/Cifrario\\_polialfabetico](https://it.wikipedia.org/wiki/Cifrario_polialfabetico)



## 22. Gioco di monete

Cristina possiede 10 monete con una faccia dorata () e una argentata ()

Cristina dispone le monete su una tavola come segue:



*Quante volte deve girare una coppia di monete vicine, in modo che alla fine mostrino tutte la faccia dorata? (Attenzione: le monete possono essere girate solo 2 alla volta!)*

- A) 1
- B) 2
- C) 4
- D) 6
- E) 8
- F) Non è possibile.



## Soluzione

Non è possibile.

Ogni volta che vengono girate due monete vicine, il numero di monete che mostrano la faccia argentata rimane sempre dispari: la “parità” delle facce delle monete non cambia.

Essendo la parità delle monete argentate sempre dispari, non mai potrà succedere che alla fine non ce ne sia almeno una.

## Questa è l'informatica!

La parità può essere calcolata facilmente in modo veloce. Essa viene impiegata per verificare la correttezza di una trasmissione, come ad esempio la lettura di un codice a barre o il numero di una fattura. Con metodi più complessi per la verifica, si possono addirittura correggere eventuali errori, senza necessità di ritrasmettere l'informazione.





## Parole chiave e siti web








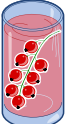



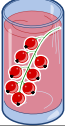



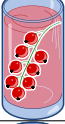
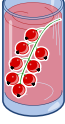



parità, bit di parità

- [https://it.wikipedia.org/wiki/Bit\\_di\\_parità](https://it.wikipedia.org/wiki/Bit_di_parità)



## 23. Succhi di frutta

Sulla strada per le vacanze, quattro amici fanno una pausa in un negozio dove è possibile acquistare rinfrescanti succhi di frutta. Ognuno dei quattro amici ha le sue preferenze, rappresentate nella tabella qui sotto. Più cuori indicano una maggiore preferenza per la bevanda. Ad esempio, Anna valuta la preferenza per la bevanda  con tre cuori e quella per la bevanda  con un cuore. Al contrario Daniel valuta la bevanda  con quattro cuori e la bevanda  con un solo cuore.

				
Anna				
Beat				
Christine				
Daniel				

I succhi di frutta sono andati a ruba e purtroppo ne rimane *solo uno* per ogni tipo. Assegna a ciascuno degli amici una bevanda diversa in modo da ottenere il maggior numero di cuori in totale.



## Soluzione

Il numero massimo di cuori ottenibile è 14, ad esempio con la soluzione seguente:

Anna				
Beat				
Christine				
Daniel				

Per ottenere questa soluzione, iniziamo da Daniel. Egli valuta la bevanda con quattro cuori, mentre tutti gli altri con solo un cuore. Se poi assegniamo a Beat o a Christine la bevanda , possiamo dare agli ultimi due (Anna e Christine, rispettivamente, Anna e Beat) la loro seconda scelta (tre cuori).

Tre dei quattro amici preferiscono la bevanda . Dato che però solo una è disponibile, due di loro dovranno necessariamente passare alla seconda scelta. Dunque non è possibile ottenere una combinazione migliore di  $3 + 3 + 4 + 4 = 14$  cuori.

Esistono solo queste due soluzioni con 14 cuori, visto che tutte le altre richiederebbero a uno a più amici di optare per la terza scelta: il massimo numero di cuori sarebbe quindi  $2 + 3 + 4 + 4 = 13$ .

## Questa è l'informatica!

In questo compito dobbiamo *ottimizzare* (qui, massimizzare) il numero di cuori e, dunque, la soddisfazione totale dei quattro amici. L'ottimizzazione è un campo di ricerca molto importante in informatica e in matematica, dato che essa si applica a molte situazioni e gli algoritmi che ne risolvono i problemi sono spesso inefficienti (ossia, utilizzano molte risorse e molto tempo).

Nel nostro caso un semplice algoritmo, che possa esaminare tutte le soluzioni (possibili o impossibili) dovrebbe verificare più di 65'000 casi. Con alcuni stratagemmi, possiamo ridurre drasticamente il numero di casi da esaminare (ci sono solo 24 soluzioni per cui valutare il numero totale di cuori). Trovare lo stratagemma giusto non è però sempre evidente.

Più concretamente, il nostro compito è una forma particolare del *problema dell'accoppiamento*: ognuno dei quattro amici deve essere associato a una bevanda distinta, massimizzando la soddisfazione totale. Problemi simili sono molto comuni nel mondo reale, ad esempio possiamo pensare alle liste di attesa per il trapianto degli organi: anche in questo caso ogni paziente riceve un organo distinto





e in più dobbiamo considerare *alcuni vincoli* (in inglese *constraint*) come tempi di attesa, urgenza e compatibilità.

## Parole chiave e siti web

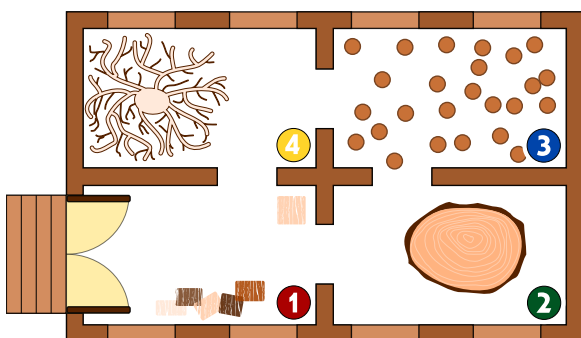
Ottimizzazione, accoppiamento (matching)

- [https://it.wikipedia.org/wiki/Ottimizzazione\\_\(matematica\)](https://it.wikipedia.org/wiki/Ottimizzazione_(matematica))
- [https://it.wikipedia.org/wiki/Branch\\_and\\_bound](https://it.wikipedia.org/wiki/Branch_and_bound)
- [https://it.wikipedia.org/wiki/Accoppiamento\\_\(teoria\\_dei\\_grafi\)](https://it.wikipedia.org/wiki/Accoppiamento_(teoria_dei_grafi))





## 24. Intrusione al museo



Nel nuovo museo di sculture in legno c'è un sistema di sicurezza intelligente. Dato che le sculture sono molto attrattive, la gente si muove con lentezza nelle varie stanze.

Ad ogni minuto, il sistema di sicurezza conta il numero di persone in ogni stanza e inserisce i dati in una tabella. Inoltre, basandosi su questa conta, controlla se nel museo c'è un intruso. Un intruso è una persona presente nel museo che però non è entrata dalla porta d'ingresso.

Se il sistema pensa che c'è un intruso, suona l'allarme.

Qui sotto trovi la tabella compilata dal sistema ogni minuto dalle 10:01 alle 10:07. Qui sopra, invece, trovi la mappa numerata delle stanze del museo.

Ora	Stanza 1	Stanza 2	Stanza 3	Stanza 4
10:01	2	0	0	0
10:02	3	0	0	0
10:03	2	1	0	0
10:04	4	1	1	0
10:05	2	2	3	0
10:06	5	2	2	1
10:07	4	1	2	2

*In quale minuto suonerà il sistema d'allarme?*

- A) 10:01
- B) 10:02
- C) 10:03
- D) 10:04
- E) 10:05
- F) 10:06
- G) 10:07



## Soluzione

L'allarme suona alle 10:05. In questo minuto nella stanza 3 ci sono due persone in più rispetto al minuto precedente, ma nelle stanze vicine alle 10:04 c'era una sola persona (stanza 2). Di conseguenza, una persona deve essere entrata nel museo senza passare dalla porta d'ingresso.

Negli altri orari non è stato individuato alcun intruso:

- Alle 10:01 ci sono 2 persone nella stanza 1, che possono essere entrate dalla porta d'ingresso.
- Dalle 10:01 alle 10:02 può essere entrata un'ulteriore persona nella stanza 1.
- Dalle 10:02 alle 10:03 una persona può essere andata dalla stanza 1 alla stanza 2.
- Dalle 10:03 alle 10:04 una persona può essere andata dalla stanza 2 alla stanza 3 e una dalla stanza 1 alla stanza 2. Inoltre 3 persone possono essere entrate dall'ingresso.
- Dalle 10:05 alle 10:06 una persona può essere andata dalla stanza 3 alla stanza 4 e 4 persone possono essere entrate dall'ingresso.
- Dalle 10:06 alle 10:07 una persona può essere andata dalla stanza 1 alla stanza 4, una dalla stanza 2 alla stanza 1 e una persona può essere uscita attraverso l'ingresso.

## Questa è l'informatica!

Sistemi di allarme che controllano il numero di persone in ambienti sensibili possono essere presenti ad esempio negli aeroporti. Programmi per computer possono riconoscere le persone dalle immagini delle telecamere nascoste, contarle e decidere di conseguenza il da farsi. Questi programmi sfruttano la cosiddetta intelligenza artificiale per riconoscere gli esseri umani ma usano anche semplici regole, come nel nostro compito, per individuare lacune nella sicurezza.

## Parole chiave e siti web

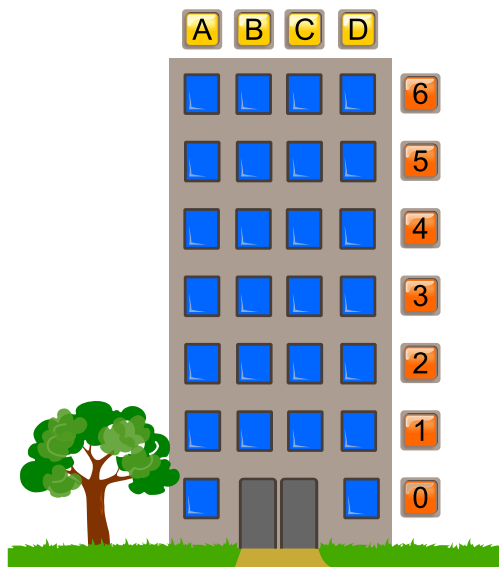
Sistema di sicurezza intelligente, regola, transizione di stato

- [https://it.wikipedia.org/wiki/Intrusion\\_detection\\_system](https://it.wikipedia.org/wiki/Intrusion_detection_system)

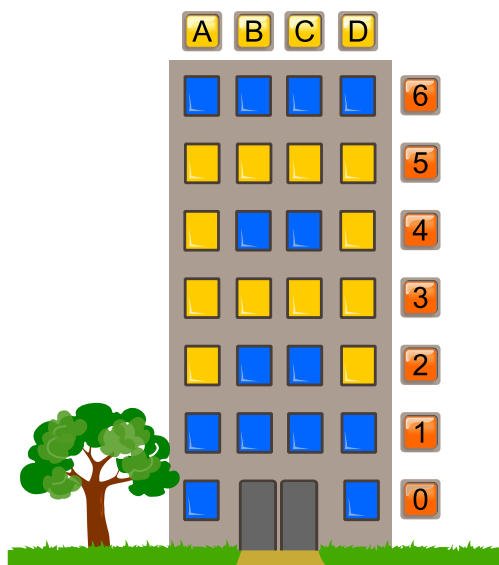


## 25. Giochi di luce

Un nuovo grattacielo in città possiede un impianto centralizzato per accendere o spegnere le luci. Il grattacielo ha 26 finestre, attraverso cui è possibile vedere se i locali sono illuminati o no. Purtroppo non è possibile accendere la luce in ogni locale singolarmente, si può solo accedere o spegnere la luce di un intero piano o un'intera colonna di finestre.



*Clicca sul numero del piano o sul nome della colonna in modo che alla fine il grattacielo appaia così:*





## Soluzione

Il compito può essere facilmente risolto accendendo dapprima i piani 3 e 5 e poi le colonne A e D. Infine bisogna spegnere i piani 6, 1 e 0. Naturalmente esistono altre soluzioni, ma tutte hanno in comune questa sequenza di operazioni.

## Questa è l'informatica!

Gli interruttori della luce in questo compito possono essere paragonati ai comandi di un qualsiasi sistema o macchina informatica. Anche i programmi di computer più complessi, infatti, possono essere interpretati come una sequenza di semplici comandi. I locali dietro alle finestre corrispondono a delle celle di memoria che possono avere valore 0 (luci spente) o 1 (luci accese).

I computer moderni permettono di eseguire comandi composti che gestiscono contemporaneamente più celle di memoria. Nel nostro caso, corrispondono agli interruttori del piano o della colonna di finestre. I computer manipolano contemporaneamente milioni di celle di memoria già solo nel processore, e diventano miliardi o bilioni nella RAM e nel disco rigido.

Per questo è importante che i comandi siano definiti chiaramente, ovvero che si specifichi quando possono essere eseguiti (*pre-condition*) e cosa succede dopo (*post-condition*). Nel nostro esempio, valgono le seguenti condizioni per le luci di un piano: se anche una sola luce del piano è spenta (*pre-condition*), dopo aver premuto l'interruttore tutte le luci del piano saranno accese (*post-condition*). Altrimenti (cioè tutte le luci sono accese, *pre-condition*) tutte le luci del piano saranno spente (*post-condition*).

## Parole chiave e siti web

Programmazione del hardware, sequenze di operazioni, operazioni binarie

- <https://it.wikipedia.org/wiki/Assembly>



## 26. Sostituzioni

Il signor Rossi si è purtroppo ammalato. Il signor Verdi deve quindi sostituirlo nei suoi incarichi nella ditta in cui lavora. Per fortuna, dopo 2 settimane il signor Rossi può tornare al lavoro. Dato, però, che il signor Verdi ha già iniziato parecchi lavori di competenza del signor Rossi, i due concordano che Verdi continuerà a lavorarci sopra, mentre Rossi assumerà i compiti originali di Verdi. La documentazione del progetto dovrà quindi sostituire il nome di Rossi con quello di Verdi e viceversa. Nella documentazione, qualsiasi testo può essere sostituito da un altro facilmente.

*Quale delle seguenti procedure farà in modo di scambiare i nomi opportunamente, ammettendo che nel testo originale non esiste il simbolo “#”?*

- A) Sostituisco dapprima tutti i “Rossi” con “Verdi” e poi tutti i “Verdi” con “Rossi”.
- B) Sostituisco dapprima tutti i “Verdi” con “Rossi” e poi tutti i “Rossi” con “Verdi”.
- C) Sostituisco dapprima tutti i “Rossi” con “#”, poi tutti i “#” con “Verdi” e infine tutti i “Verdi” con “Rossi”.
- D) Sostituisco dapprima tutti i “Rossi” con “#”, poi tutti i “Verdi” con “Rossi” e infine tutti i “#” con “Verdi”.



## Soluzione

La risposta corretta è D).

- A) Alla fine troveremmo solo il nome “Rossi” e non vi sarebbe traccia di “Verdi”, poiché con l’ultima operazione tutti i “Verdi” verrebbero rimpiazzati da “Rossi”.
- B) Alla fine troveremmo solo il nome “Verdi” e non vi sarebbe traccia di “Rossi”, poiché con l’ultima operazione tutti i “Rossi” verrebbero rimpiazzati da “Verdi”.
- C) Alla fine troveremmo solo il nome “Rossi” e non vi sarebbe traccia di “Verdi”, poiché con l’ultima operazione tutti i “Verdi” verrebbero rimpiazzati da “Rossi”.
- D) È l’unico procedimento valido. I “Rossi” vengono dapprima temporaneamente sostituiti da “#”, in modo da poter scambiare i vecchi incarichi di “Verdi” con “Rossi”. A questo punto possiamo assegnare i vecchi incarichi di “Rossi”, salvati come “’#”, a “Verdi”.

## Questa è l’informatica!

Sebbene la procedura di scambio sia molto semplice, in informatica essa possiede un grande importanza. Grazie a questa procedura, infatti, possono essere eseguiti dei compiti molto complessi. Molte grammatiche formali (alla base dei linguaggi di programmazione) sono definite come una lista di regole di sostituzione.

In questo compito vediamo come sia necessario disporre di un elemento “temporaneo” per poter eseguire lo scambio di due valori: questo concetto è importantissimo quando si parla di scambio di variabili (swap).

## Parole chiave e siti web

elaborazione di testi, sequenze di istruzioni, scambio (swap) di variabili






- [https://it.wikipedia.org/wiki/Grammatica\\_formale](https://it.wikipedia.org/wiki/Grammatica_formale)





## 27. Esci dal labirinto!

Luca vuole attraversare un labirinto. Non sapendo come fare, ti chiede di aiutarlo a trovare l'uscita. Egli entra nel labirinto nella posizione indicata dal triangolo nero e vuole raggiungere l'uscita indicata dal cerchio rosso. Luca può, però, ricordarsi solo 8 delle seguenti istruzioni:

		Fai un passo in avanti e girati a sinistra.
		Fai un passo in avanti e girati a destra.
		Fai un passo in avanti.

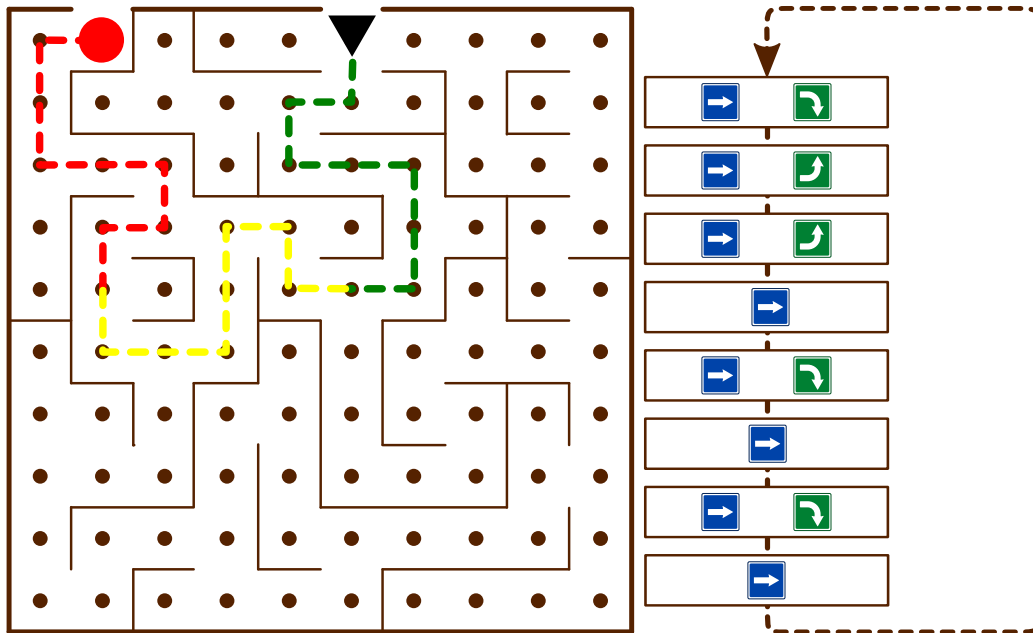
Anche se Luca può ricordarsi solo un massimo di otto istruzioni, può comunque ripeterle in sequenza più volte fino all'uscita.

All'inizio Luca si trova nella posizione indicata dal triangolo nero, rivolto verso il basso. Seleziona le istruzioni e scrivile nella sequenza corretta nei campi vuoti che trovi qui sotto, in modo che Luca possa giungere all'uscita.



## Soluzione

La seguente sequenza di istruzioni porterà Luca all'uscita, se ripetuta per tre volte:



## Questa è l'informatica!

Luca esegue di fatto un programma. Questo programma è composto da una *sequenza* di istruzioni. Una *struttura di controllo* come il *ciclo* (*loop*) in questo programma permette di ripetere una sequenza di istruzioni più volte fino a quando necessario. In questo modo possiamo evitare di ricopiare tale sequenza nel codice del programma e dunque risparmiare tempo. Inoltre è anche più facile individuare e correggere eventuali errori.

## Parole chiave e siti web

Sequenza di istruzioni, cicli (ripetizioni, "loop"), algoritmi

- [https://it.wikipedia.org/wiki/Struttura\\_di\\_controllo](https://it.wikipedia.org/wiki/Struttura_di_controllo)

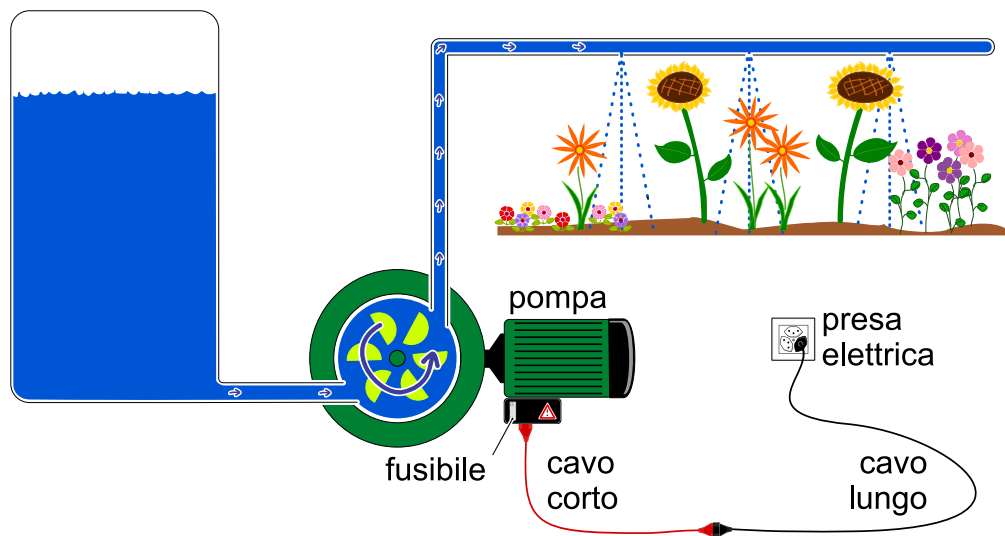


## 28. Impianto di irrigazione

Gianni possiede un giardino fiorito e un orto, per i quali ha costruito due impianti di irrigazione identici. Qui sotto possiamo vedere lo schema di uno di essi.

L'impianto è collegato alla rete elettrica attraverso una serie di componenti:

- un cavo lungo
- un cavo corto
- una pompa
- un fusibile, che impedisce il funzionamento della pompa quando salta (“si rompe”).



Un giorno Gianni scopre che l'impianto di irrigazione del giardino fiorito non funziona. Egli riesce a capire che il problema non risiede né nel serbatoio, né nelle tubazioni dell'acqua.

Solo un componente è difettoso e Gianni vuole identificarlo. Per questo compito può utilizzare tutti i pezzi dell'impianto di irrigazione dell'orto. Scambiando le diverse parti, è sicuro di poter trovare il componente difettoso.

*Indica quali delle seguenti affermazioni sono corrette.*

- Il primo componente da testare è di sicuro la pompa, perché essa è la cosa più importante.
- Gianni può impiegare un qualsiasi numero di componenti del secondo impianto di irrigazione, ma per ogni passo un solo componente deve essere sostituito. Se il primo impianto torna a funzionare, allora è l'ultimo componente sostituito quello difettoso.
- Prima di tutto si può controllare la presa elettrica con uno strumento apposito, per verificare se c'è corrente. In caso affermativo, si può procedere passo per passo sostituendo una componente dopo l'altra.
- Si devono comprare dei componenti nuovi: quelli dell'impianto dell'orto sono già stati usati!
- Ad ogni passo devono essere sostituiti almeno 2 componenti. In questo modo è possibile giungere alla soluzione più velocemente.



## Soluzione

Le risposte corrette sono B) e C).

Risposta A): non importa quanto un componente possa essere reputato importante, questo non ci dice nulla sulla sua probabilità di rottura.

Risposta D): Anche i pezzi nuovi possono essere talvolta difettosi. I componenti dell'impianto dell'orto, invece, funzionano sicuramente.

Risposta E): Il sistema sarebbe di sicuro riparato più velocemente, basterebbe sostituire ogni componente dell'impianto difettoso con quello dell'impianto funzionante. Gianni desidera però identificare il componente difettoso e con questo metodo non è possibile farlo.

## Questa è l'informatica!

La ricerca delle cause di un errore in un'applicazione è un'attività importante del cosiddetto "*Debugging*".

Trovare la causa di un singolo errore è spesso già una sfida molto complessa, ma trovare le cause di più errori concomitanti è difficilissimo. Per questo gli sviluppatori di software hanno adottato una strategia che consiste nel testare un'applicazione (possibilmente in modo automatico) dopo ogni singolo cambiamento nel codice. In questo modo si aumentano le probabilità che si debba cercare una sola causa per un eventuale errore.

## Parole chiave e siti web

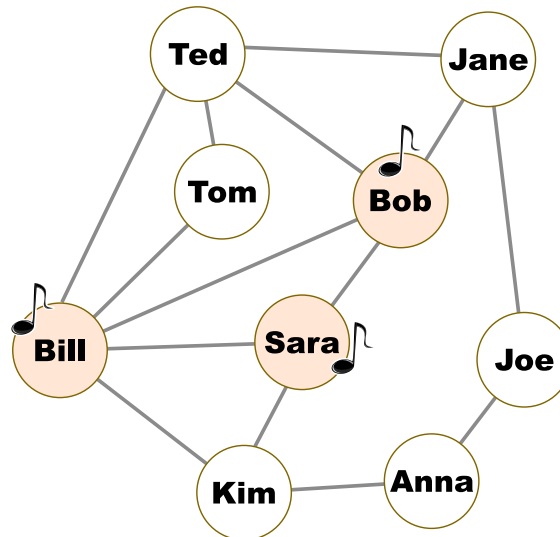
Debugging, verifica del software

- <https://en.wikipedia.org/wiki/Systemantics>
- [https://en.wikiquote.org/wiki/John\\_Gall](https://en.wikiquote.org/wiki/John_Gall)
- <https://it.wikipedia.org/wiki/Debugging>



## 29. Una nuova canzone

Nel diagramma seguente due persone sono amiche se i loro nomi sono uniti da una linea; in caso contrario, non lo sono. Lunedì una Pop Star ha pubblicato una nuova canzone. Lo stesso giorno Bill, Bob e Sara la hanno acquistata. I loro nomi sono dunque annotati con una nota musicale.



A partire da Martedì, ogni giorno, succede questo: ogni persona del gruppo acquista la canzone se il giorno precedente almeno la metà dei suoi amici l'aveva acquistata. Martedì, ad esempio, Tom acquista la canzone, ma Jane no.

*In quale giorno tutte le persone del gruppo hanno acquistato la canzone?*

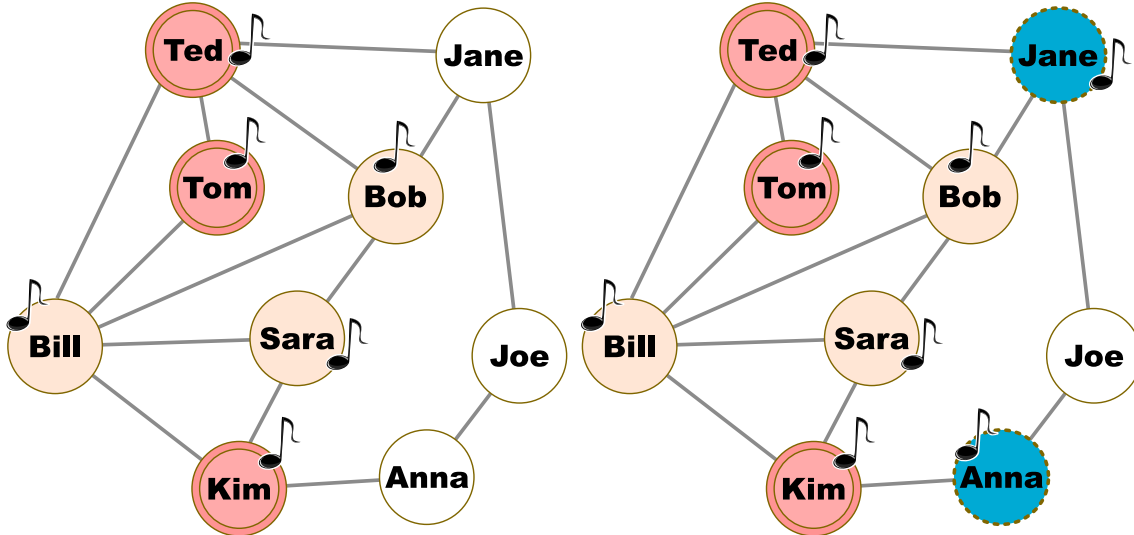
- A) Mercoledì
- B) Giovedì
- C) Venerdì
- D) Sabato



## Soluzione

La risposta corretta è B).

Tom, Ted e Kim comprano la canzone Martedì. Anna e Jane Mercoledì:



Infine, Joe acquista la canzone Giovedì. Dunque Giovedì tutte le persone del gruppo hanno acquistato la canzone, non prima.

## Questa è l'informatica!

I social network hanno spesso più di un miliardo di membri. Per questo, oltre che per gli utenti, essi sono molto appetibili anche per le società di marketing. In molte campagne pubblicitarie vengono infatti sfruttati i social network per diffondere le informazioni su un nuovo prodotto. Questo processo è alla base anche del nostro compito.

Qui viene simulata la diffusione con un modello detto “a soglia”. Il diagramma è di fatto un grafo formato da nodi (i nomi di persona) e archi (le linee che indicano la relazione di amicizia). I nodi collegati da archi sono detti “vicini”. La soglia di “infezione”  $q$  è un valore tra 0 e 1. Nel nostro compito  $q = 0.5$ : una persona mostra un nuovo comportamento B (qui, compra la canzone), quando almeno una parte pari a  $q$  dei vicini ha assunto tale comportamento B.

## Parole chiave e siti web

Social network, grafo

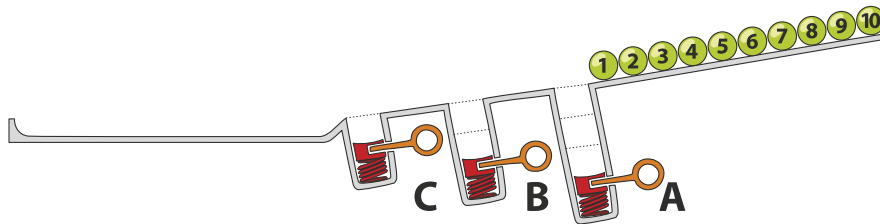
- [https://it.wikipedia.org/wiki/Teoria\\_del\\_mondo\\_piccolo](https://it.wikipedia.org/wiki/Teoria_del_mondo_piccolo)



## 30. Pista per biglie

Sopra una rampa sono allineate 10 biglie numerate. Lungo la rampa sono disposte tre buche A, B e C: la buca A può contenere tre biglie, la buca B due biglie e la buca C una sola biglia. Quando le biglie rotolano lungo la rampa cadono dapprima nelle buche fino a riempirle (dunque le biglie 1, 2 e 3 nella buca A, le biglie 4 e 5 nella buca B e la biglia 6 nella buca C), mentre le rimanenti scorrono via.

Quando tutte le biglie sono scese, vengono liberate le buche precedentemente riempite: dapprima la buca A, poi la buca B e infine la buca C. Prima di rilasciare la molla di una buca, si attende che le altre biglie siano passate.



*In quale sequenza troveremo le biglie alla fine?*

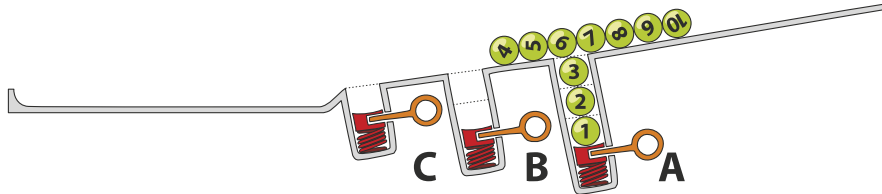
- A)  B)  C)  D) 



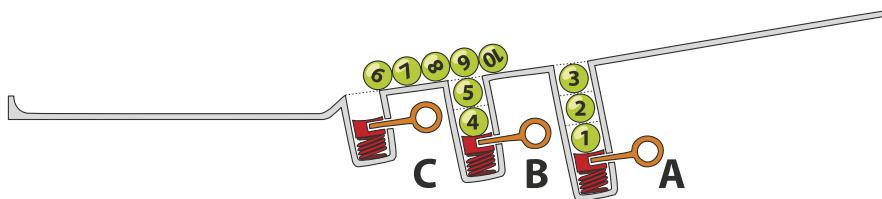
## Soluzione

La risposta corretta è D): 7 8 9 10 3 2 1 5 4 6.

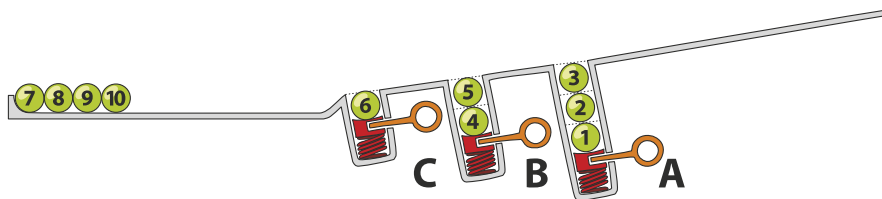
Le biglie 1, 2 e 3 cadono nella buca A.



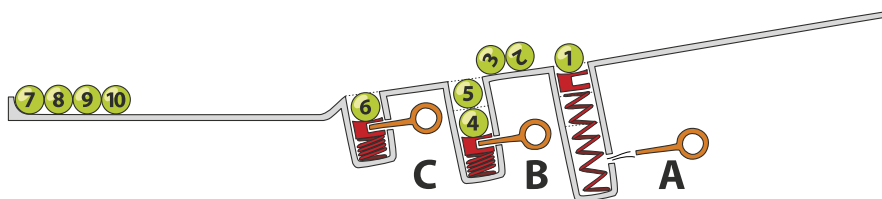
Le biglie 4 e 5 superano la buca A (ormai piena) e cadono nella buca B.



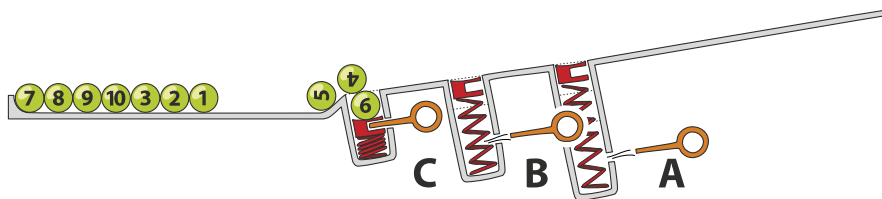
La biglia 6 cade nella buca C. Le biglie da 7 a 10 giungono invece alla fine della rampa.



Dopo il rilascio della molla nella buca A, le biglie 3, 2 e 1 arrivano alla fine della rampa in ordine invertito rispetto alla partenza. La sequenza temporanea è quindi 7, 8, 9, 10, 3, 2, 1.

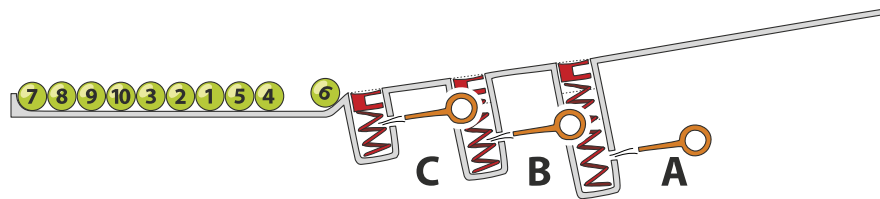


Dopo il rilascio della molla nella buca B, anche le biglie 5 e 4 giungono alla fine della rampa.



Infine, anche la buca C viene liberata permettendo così alla biglia 6 di chiudere la fila. La sequenza finale è perciò 7, 8, 9, 10, 3, 2, 1, 5, 4, 6.





## Questa è l'informatica!

Le buche della rampa ricordano le strutture di dati chiamate pile (in inglese, *stack*). Una pila permette di salvare e poi utilizzare dei dati secondo il principio *Last-In First-Out (LIFO)*: l'ultima biglia ad entrare in una buca è anche la prima ad uscire. Sebbene questo principio possa sembrare molto semplice, esso è utilizzato in molte situazioni. Per esempio possiamo verificare se le parentesi in un'espressione aritmetica sono bilanciate oppure no. Nell'espressione  $((1 + 2) \cdot 3)$  le parentesi sono bilanciate, mentre in  $((4 + 5) \cdot (6 - 7))$  no. Ogni parentesi aperta viene inserita nella pila (con l'operazione chiamata *push*) e quando si incontra una parentesi chiusa allora la si estrae (con l'operazione chiamata *pop*). Se non ci sono più parentesi aperte da estrarre, oppure se alla fine dell'espressione la pila contiene ancora della parentesi aperte, allora significa che c'è un errore di bilanciamento. Al contrario, se alla fine dell'espressione la pila è vuota, significa che l'espressione è ben bilanciata.

## Parole chiave e siti web

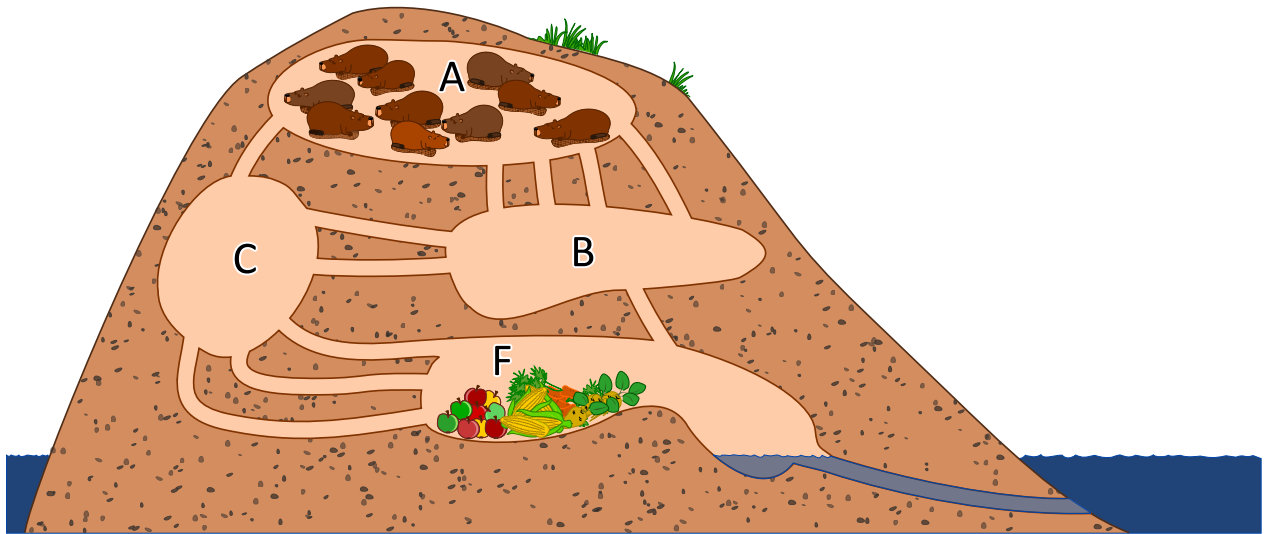
stack (pila), LIFO

- <https://it.wikipedia.org/wiki/LIFO>
- [https://it.wikipedia.org/wiki/Pila\\_\(informatica\)](https://it.wikipedia.org/wiki/Pila_(informatica))





## 31. I tunnel della diga



10 castori si trovano nella stanza A e vogliono arrivare velocemente nella stanza F per mangiare. Ogni castoro impiega 1 minuto a percorrere un tunnel di collegamento tra due stanze. Purtroppo, però, un tunnel può essere percorso da un solo castoro alla volta. Non è dunque possibile che un tunnel sia percorso da due castori contemporaneamente. Nelle stanze A, B, C ed F c'è abbastanza spazio per ospitare un numero qualsiasi di castori e non viene impiegato tempo per attraversarle. *Dopo quanti minuti possiamo già trovare tutti i castori nella stanza F? Indica il tempo minimo in assoluto.*



## Soluzione

Tutti i 10 castori possono già trovarsi nella stanza F in esattamente 4 minuti.

Nella diga esistono due cammini minimi verso la stanza F. Entrambi possono essere percorsi in 2 minuti da un castoro ciascuno:

- $A \rightarrow B \rightarrow F$
- $A \rightarrow C \rightarrow F$

Dopo 2 minuti troviamo quindi un massimo di 2 castori nella stanza F, mentre in 3 minuti possono giungervi altri 2 castori da questi percorsi.

Il percorso  $A \rightarrow B \rightarrow C \rightarrow F$  ha una capacità per 2 castori, ma impiega 3 minuti. Dopo 3 minuti possiamo trovare dunque fino a 6 castori nella stanza F (4 attraverso i percorsi minimi e 2 da quello più lungo). Al quarto minuto tutti i castori giungono alla stanza F. La tabella seguente mostra in dettaglio ciò che succede:

Azione / Situazione	Numero di castori nella stanza (dopo l'azione)			
	A	B	C	F
Situazione iniziale	10	0	0	0
<i>3 castori vanno da A a B (non sfruttando tutti i tunnel)</i> <i>1 castoro va da A a C</i>				
Situazione dopo 1 minuto	6	3	1	0
<i>3 castori vanno da A a B (non sfruttando tutti i tunnel)</i> <i>1 castoro va da B a F</i> <i>2 castori vanno da B a C</i> <i>1 castoro va da C a F</i> <i>1 castoro va da A a C</i>				
Situazione dopo 2 minuti	2	3	3	2
<i>1 castoro va da A a B (cammino minimo)</i> <i>1 castoro va da B a F</i> <i>2 castori vanno da B a C</i> <i>1 castoro va da A a C (cammino minimo)</i> <i>3 castori vanno da C a F</i>				
Situazione dopo 3 minuti	0	1	3	6
<i>1 castoro va da B a F</i> <i>3 castori vanno da C a F</i>				
Situazione dopo 4 minuti	0	0	0	10

Esistono altre possibili soluzioni che consentono a tutti i castori di arrivare alla stanza F entro 4 minuti. In quella mostrata, nessun castoro deve attendere nelle stanze intermedie prima di poter proseguire.

## Questa è l'informatica!

La rete di tunnel della diga può essere rappresentata attraverso una cosiddetta rete di flusso. Il numero di tunnel tra due stanze determina quanti castori possono passare in un minuto. Questa caratteristica è la *capacità della connessione* tra due stanze, e dunque rappresenta il flusso massimo tra di esse.



Nella teoria dei grafi, una rete di flusso è modellata attraverso un grafo orientato, nel quale gli archi possiedono una capacità massima. Un flusso, che scorre attraverso gli archi, è limitato da questa capacità. Con l'aiuto delle reti di flusso è possibile simulare reti telematiche o reti stradali, riuscendo a scoprire quali punti rappresentino dei colli di bottiglia e determinino quindi degli intasamenti. Nelle reti di flusso, un dato particolarmente interessante è il flusso massimo ottenibile tra due nodi. Nel nostro compito solo 4 castori possono passare dalla stanza A alla stanza F ad ogni minuto, senza dover attendere nelle stanze intermedie. Questo flusso massimo può essere calcolato facilmente con l'algoritmo di Ford-Fulkerson.

## Parole chiave e siti web





grafo, flusso di rete

- [https://it.wikipedia.org/wiki/Rete\\_di\\_flusso](https://it.wikipedia.org/wiki/Rete_di_flusso)
- [https://it.wikipedia.org/wiki/Digrafo\\_\(matematica\)](https://it.wikipedia.org/wiki/Digrafo_(matematica))
- [https://it.wikipedia.org/wiki/Algoritmo\\_di\\_Ford-Fulkerson](https://it.wikipedia.org/wiki/Algoritmo_di_Ford-Fulkerson)

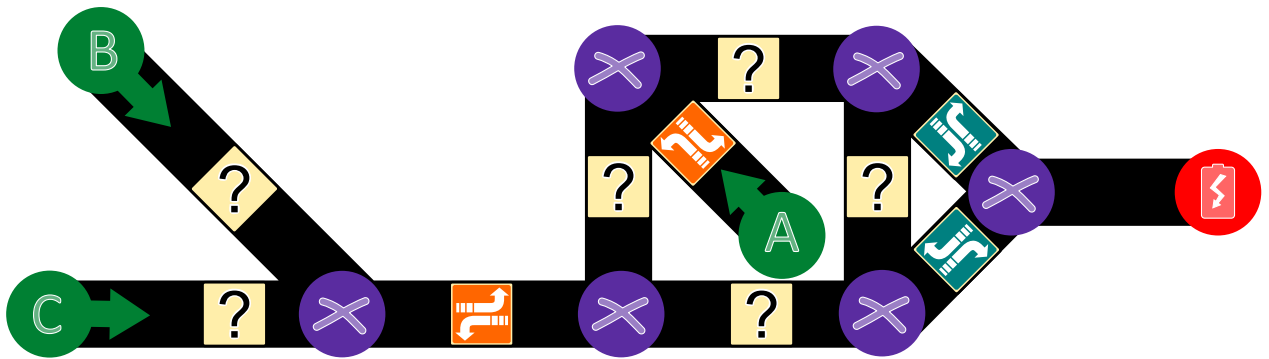




## 32. Aiuta l'Arabot

Un Arabot è un robot che si muove su un foglio di carta. Esso segue sempre le linee nere che vengono disegnate sul foglio. Su ogni linea si trova un'indicazione che gli dice se al prossimo incrocio () deve prendere la prima svolta alla sua sinistra () o la prima svolta alla sua destra (). Se dovesse giungere ai punti A, B o C egli non saprebbe cosa fare e quindi si spegnerebbe. L'Arabot può partire per il suo cammino solo dai punti A, B o C e alla fine deve sempre arrivare alla stazione di ricarica ().

Jonas ha disegnato un percorso per proprio robot. Egli ha pure già inserito alcune indicazioni, ma per alcuni punti non sa cosa scegliere.

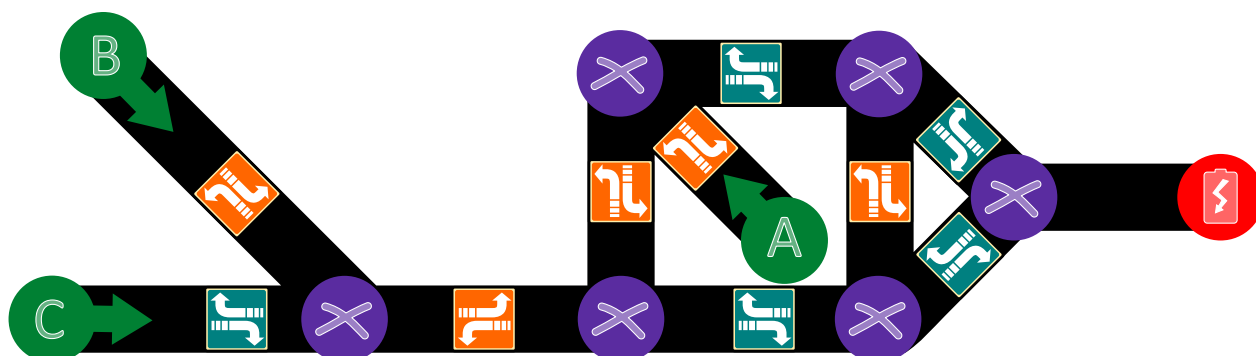


*Aiuta Jonas a trovare le giuste indicazioni per tutte le linee.*



## Soluzione

La risposta corretta è:



Per entrambe le linee più a sinistra che partono da B e C, dobbiamo assicurarci che il robot possa proseguire verso la linea a destra e dunque sopra deve seguire l'indicazione mentre sotto quella .

Nella parte destra del percorso, seguendo il cammino da B e C, la prima linea verticale deve poter condurre alla stazione di ricarica e quindi dobbiamo dare l'indicazione , altrimenti si finirebbe in A. La linea orizzontale in alto, invece, deve indicare necessariamente , altrimenti in nessun modo il percorso fin qui seguito potrebbe mai condurre a destinazione. Analogamente la linea verticale di destra deve avere l'indicazione . Infine, seguendo il percorso da A, vediamo come la linea alla base del “quadrato” debba avere l'indicazione .

## Questa è l'informatica!

Questo compito consiste nel codificare dei cammini diversi verso una certa destinazione (da A a , da B a o da C a ) con una determinata struttura (nel nostro caso un grafo). In informatica essa è detta struttura dei dati. Quando l'Arabot segue un certo percorso (ad esempio da A a , esso deve leggere ed eseguire determinate istruzioni: “In quale direzione devo svoltare al prossimo incrocio? Se mi viene indicato, prenderò quella strada!”. Un computer funziona dal punto di vista del hardware allo stesso modo: legge istruzioni e le esegue.

Con questo esercizio possiamo intuire molte domande interessanti dal punto di vista matematico e informatico. Ad esempio: quanto è difficile dare delle istruzioni corrette e univoche per eseguire un determinato compito? Molte domande simili sono tutt'ora senza risposta e costituiscono uno degli obiettivi nel campo di ricerca dell'*algoritmica* e della *teoria della complessità*. Competenze simili possono essere applicate anche nel campo della *medicina* e della *biologia computazionale*.

## Parole chiave e siti web

Grafo bidirezionale, teoria della complessità, biologia computazionale, medicina computazionale

- [https://it.wikipedia.org/wiki/Teoria\\_della\\_complessità\\_computazionale](https://it.wikipedia.org/wiki/Teoria_della_complessità_computazionale)
- [https://en.wikipedia.org/wiki/Computational\\_biology](https://en.wikipedia.org/wiki/Computational_biology)
- [https://en.wikipedia.org/wiki/In\\_silico\\_medicine](https://en.wikipedia.org/wiki/In_silico_medicine)





## 33. Gioco con stuzzicadenti

Lucia e Marco si divertono con un gioco basato su degli stuzzicadenti. All'inizio due mucchi di stuzzicadenti sono disposti su un tavolo. Ad ogni turno un giocatore...

1. ...elimina uno dei due mucchi di stuzzicadenti...
2. ...e divide l'altro in due mucchi.

Un giocatore vince se lascia sul tavolo due "mucchi" composti da un solo stuzzicadenti ciascuno. Inizia Lucia...

*Lucia inizia con 24 stuzzicadenti a disposizione, da suddividere in due mucchi. Scegli le suddivisioni per i due mucchi che consentano a Lucia di vincere:*

- A) 11 e 13
- B) 12 e 12
- C) 7 e 17
- D) 8 e 16



## Soluzione

Lucia deve suddividere gli stuzzicadenti in due mucchi da 11 e 13 oppure 7 e 17 per poter vincere. Per poter vincere Lucia deve scegliere una suddivisione iniziale che crei due mucchi composti da un numero dispari di stuzzicadenti. In caso contrario, concederebbe la possibilità di vincere a Marco. . . Perché deve adottare questa strategia? Se un giocatore lascia due mucchi “dispari” di stuzzicadenti, il prossimo giocatore potrà solo lasciarne due con un numero dispari e uno pari di stuzzicadenti. Il primo giocatore allontanerà quindi il mucchio “dispari” e suddividerà il rimanente in due mucchi “dispari”. Il gioco finisce quando rimangono due mucchi composti da un solo stuzzicadenti . . . dunque due mucchi dispari. Quindi può solo vincere il giocatore che lascia 2 mucchi “dispari”.

## Questa è l'informatica!

Le strategie vincenti per giochi simili a quello proposto sono semplici da trovare: è sufficiente identificare un'invariante (ovvero una proprietà che non cambia durante la partita) che possa condurre alla vittoria. In questo caso, l'invariante consiste nel lasciare 2 mucchi “dispari” di stuzzicadenti.

Nel nostro gioco, non è solo necessario applicare la giusta strategia, bisogna anche impedire che l'avversario la usi. Per questo la situazione di partenza (numero degli stuzzicadenti) e la scelta di chi debba iniziare, spesso ne determina l'esito, se tutti si comportano in modo “ottimale”.

Gli informatici si occupano spesso di giochi simili, in cui nulla è lasciato al caso e solo la strategia determina chi vince e chi perde. Da piccoli giochi come questo, in cui si può calcolare la mossa giusta in pochissimi secondi, a quelli più complessi come gli scacchi o Go, in cui anche i computer più avanzati non riescono a trovare la miglior mossa neppure in anni di calcoli, si impara come applicare le decisioni più corrette anche in situazioni complesse . . . sia che si tratti di giochi oppure di intelligenza artificiale.

## Parole chiave e siti web

gioco di strategia, albero di decisione, gioco in forma estesa

- [https://it.wikipedia.org/wiki/Gioco\\_astratto](https://it.wikipedia.org/wiki/Gioco_astratto)
- [https://it.wikipedia.org/wiki/Gioco\\_in\\_forma\\_estesa](https://it.wikipedia.org/wiki/Gioco_in_forma_estesa)
- [https://en.wikipedia.org/wiki/Combinatorial\\_game\\_theory](https://en.wikipedia.org/wiki/Combinatorial_game_theory)



## 34. La distanza tra parole

Per calcolare la distanza tra due parole, si considerano le seguenti operazioni:

- Inserire una lettera in un punto qualsiasi di una parola
- Eliminare una lettera da un punto qualsiasi di una parola
- Sostituire una lettera di una parola con un'altra qualsiasi

La distanza tra due parole è il numero minimo di operazioni simili, necessario per trasformare la prima parola nella seconda.

La distanza tra “cantare” e “stonare” è 4:

1. cantare → canare (eliminiamo la “t”)
2. canare → conare (sostituiamo la “a” con la “o”)
3. conare → tonare (sostituiamo la “c” con la “t”)
4. tonare → stonare (inseriamo la “s”)

Qual è la distanza tra “Emil” ed “Erich”?



## Soluzione

La distanza tra “Emil” ed “Erich” è 3, come dimostrato attraverso le seguenti operazioni:

Emil → Eril → Eric → Erich

Non è possibile compiere meno operazioni, poiché Erich possiede una lettera in più (un’operazione) e non possiede né una “m”, né una “l” (2 operazioni).

## Questa è l’informatica!

La distanza tra parole così calcolata è detta *distanza di Levenshtein*, dal ricercatore russo Vladimir Levenshtein che per primo l’ha descritta nel 1965. Essa viene utilizzata, ad esempio, per suggerire correzioni negli errori ortografici: se tale distanza tra la parola non presente nel vocabolario e una parola presente è piccola, allora molto probabilmente è stato commesso un errore di digitazione e l’editore di testo suggerisce la parola giusta. Tale distanza è usata anche per calcolare la somiglianza di stringhe di DNA, immagini o anche per fornire traduzioni automatiche di testi.

È possibile calcolare la distanza di Levenshtein con il computer, provando tutte le possibili operazioni. Per evitare che il computer si perda in parole inutilmente lunghe, vengono impartiti dei confini precisi nel quale il programma può operare.

## Parole chiave e siti web

distanza di Levenshtein, distanza “di modifica” (edit distance)

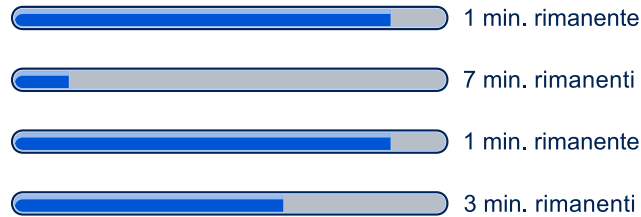
- [https://it.wikipedia.org/wiki/Distanza\\_di\\_Levenshtein](https://it.wikipedia.org/wiki/Distanza_di_Levenshtein)
- [https://en.wikibooks.org/wiki/Algorithm\\_Implementation/Strings/Levenshtein\\_distance](https://en.wikibooks.org/wiki/Algorithm_Implementation/Strings/Levenshtein_distance)



## 35. Download contemporanei

Quando si scaricano contemporaneamente più documenti di grosse dimensioni, la capacità della connessione viene condivisa. Ad esempio, se si scaricano 10 documenti, solo un decimo della velocità di connessione è disponibile per ognuno di essi.

Un utente sta scaricando 4 documenti contemporaneamente. Il tempo rimanente viene calcolato in base alla velocità attuale.



*Quanti minuti ci vogliono per completare il download dei 4 documenti.*



## Soluzione

In totale ci vogliono 3 minuti per completare il download.

Dopo 1 minuto, 2 documenti sono stati scaricati completamente. Per gli altri mancano ancora, rispettivamente, 6 e 2 minuti. La velocità di download viene però raddoppiata, in quanto i documenti rimanenti possono utilizzare il doppio della capacità. Rimangono, quindi, 3 e 1 minuto per completare il download:



Dopo un ulteriore minuto anche il terzo documento è completato. All'ultimo rimangono ancora 2 minuti, la capacità di connessione viene però raddoppiata e dunque in realtà manca solo 1 minuto:



Al terzo minuto, tutti i documenti sono stati scaricati.

## Questa è l'informatica!

L'*indicatore di avanzamento* (o *progress bar*) è un elemento informativo che mostra il progresso nell'elaborazione di un ordine. Esso è utilizzato ad esempio durante le installazioni i programmi o il download di documenti. Di regola, esso fornisce anche una stima del tempo rimanente. In casi particolari, quando non è possibile fornire una stima, ma si vuole segnalare che il computer sta elaborando ugualmente la richiesta si mostra un simbolo ruotante come una clessidra.

La velocità di trasmissione indica la quantità di dati digitali che viene trasmessa per ogni secondo. La velocità massima è detta capacità di un canale. Per ogni canale esistono però anche altre misure importanti, come ad esempio il ritardo nella risposta (o latenza). Essa indica il tempo impiegato da un dato per passare dal trasmettitore al ricevente (a volte in questa misura è calcolato anche il tempo per il percorso inverso).

Nella rete ci sono diversi fattori che influenzano la velocità massima di trasferimento: spesso il server a cui ci si connette deve far fronte a molte richieste e la sua capacità per ogni comunicazione è quindi notevolmente ridotta da limitazioni hardware o software. I dati, poi, possono arrivarci da qualsiasi parte nel mondo e devono passare attraverso numerosi punti di instradamento (router), che gestiscono molte connessioni. La velocità è pure limitata dall'allacciamento al provider di internet attraverso il modem casalingo (di norma un decimo o anche meno della velocità massima di un server). Infine, la connessione tra modem e computer (spesso attraverso un router domestico wireless) può costituire un ennesimo collo di bottiglia. La capacità di una rete WLAN, ad esempio, è molto ridotta (più o meno un decimo) rispetto alla connessione tramite cavo "ethernet".

## Parole chiave e siti web

download, indicatore di avanzamento (progress bar)

- [https://it.wikipedia.org/wiki/Progress\\_bar](https://it.wikipedia.org/wiki/Progress_bar)
- [https://de.wikipedia.org/wiki/Datenübertragungsrate#Beispiele\\_für\\_Datenübertragungsraten](https://de.wikipedia.org/wiki/Datenübertragungsrate#Beispiele_für_Datenübertragungsraten)



## 36. Raccolta di punti

Il seguente rompicapo è diventato molto popolare. Si prende una tabella come quella mostrata qui sotto. Si parte dalla casella P (per partenza) e si arriva alla casella A (per arrivo). Si procede di casella in casella, ma ad ogni passo si può solo andare verso destra o verso l'alto, come indicato dalle frecce. L'obiettivo è raccogliere più punti possibili: ogni casella indica il proprio punteggio. Bisogna quindi scegliere la strada giusta tra partenza e arrivo in modo da sommare il maggior numero di punti.

	2	0	1	1	<b>A</b>
	1	2	0	2	3
	2	2	0	2	1
	3	1	0	2	0
↑	<b>P</b>	0	1	3	0
				→	

*Qual è il punteggio massimo ottenibile?*

- A) 10
- B) 12
- C) 14
- D) 16



## Soluzione

La risposta corretta è C) 14.

Un possibile metodo per la soluzione consiste nel creare una tabella che indichi il punteggio massimo complessivo ottenibile in ogni casella. Si parte dalla casella in basso a sinistra, il cui punteggio è 0:

2	0	1	1	A
1	2	0	2	3
2	2	0	2	1
3	1	0	2	0
<b>0</b>	0	1	3	0

In essa rappresentiamo in grassetto il punteggio massimo ottenibile. Procedendo verso l'alto questo numero verrebbe incrementato di 3, a destra, invece, di 0.

2	0	1	1	A
1	2	0	2	3
2	2	0	2	1
<b>3</b>	1	0	2	0
<b>0</b>	<b>0</b>	1	3	0

La casella a destra di quella indicata con il 3 in grassetto e sopra a quella indicata con lo 0 in grassetto può essere raggiunta solo dal basso o da sinistra. Il punteggio massimo ottenibile è dunque 4 (0 + 3 + 1).

2	0	1	1	A
1	2	0	2	3
2	2	0	2	1
<b>3</b>	<b>4</b>	0	2	0
<b>0</b>	<b>0</b>	1	3	0

Procedendo in modo analogo, si può calcolare per ogni casella il punteggio massimo ottenibile, sommando il massimo tra il valore della casella a sinistra e quella sotto al valore della casella stessa. Matematicamente abbiamo l'espressione:

$$v(i, 0) = 0$$

$$v(0, j) = 0$$

$$v(i, j) = c(i, j) + \max\{v(i - 1, j), v(i, j - 1)\}$$

dove,  $v(i, j)$  è il massimo delle caselle  $(i, j)$  e  $c(i, j)$  è il valore iniziale della casella. Dato che la formula utilizza sempre le caselle a sinistra e quelle sotto, è necessario inizializzare una riga sotto e una colonna a sinistra della tabella con degli zero. Se applichiamo la formula alla nostra tabella, otteniamo:

0	8	9	10	12	14
0	6	9	9	11	14
0	5	7	7	9	10
0	3	4	4	6	6
0	0	0	1	4	4
0	0	0	0	0	0





## Questa è l'informatica!

Trovare la soluzione migliore nell'insieme di tutte le soluzioni possibili richiede spesso stratagemmi elaborati e non evidenti. Nel nostro problema, potremmo provare tutti i cammini possibili, adottando un metodo chiamato “forza bruta”, ma ciò che ci costringerebbe a valutare ben 70 diverse soluzioni. Un metodo più efficiente, come mostrato nella nostra spiegazione della risposta, consiste nel annotarsi delle soluzioni parziali. Questa tecnica è detta *memoizzazione* (la mancanza della ‘r’ non è un errore di battitura!) e fa in modo di salvare i risultati parziali in memoria permettendo l'applicazione della programmazione dinamica (divisione di un problema in tanti sotto-problemi). In questo modo dobbiamo calcolare solo 25 valori.

Nel nostro caso, comunque, possiamo anche trovare dei cammini parziali redditizi, dai cui è possibile giungere velocemente alla miglior soluzione. Dato che la tabella è molto piccola, è anche semplice verificare che le altre soluzioni sono peggiori.

## Parole chiave e siti web

Programmazione dinamica, Memoizzazione

- [https://it.wikipedia.org/wiki/Programmazione\\_dinamica](https://it.wikipedia.org/wiki/Programmazione_dinamica)
- <https://it.wikipedia.org/wiki/Memoizzazione>





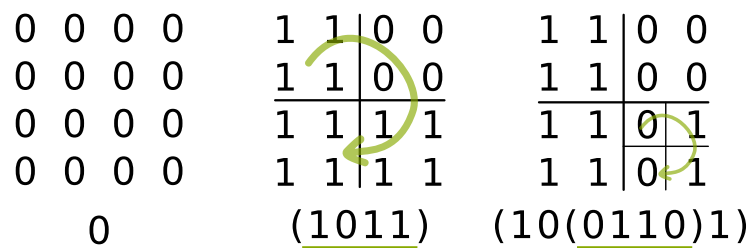
## 37. Metodo dei quarti

Immagini di pixel in bianco e nero possono essere rappresentate da simboli binari: 0 descrive un pixel bianco, 1 descrive un pixel nero. Un'immagine di  $4 \times 4$  pixel è dunque rappresentata da 16 simboli.

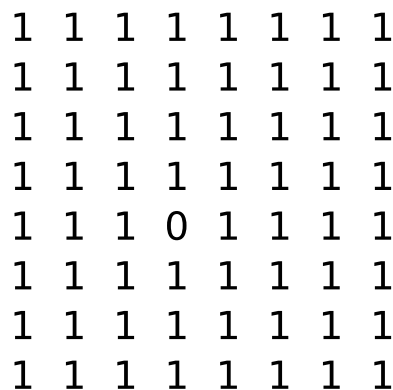
Molte immagini possono però essere rappresentate da una minore quantità di simboli, quando viene applicato il metodo dei "quarti". Con esso, i simboli binari dell'immagine vengono collocati in su una griglia quadrata. Il metodo dei quarti viene poi applicato come segue:

Se la griglia è composta solo dallo stesso simbolo binario, allora il risultato è tale simbolo. In altre parole, se tutti i simboli di una griglia sono 0, allora il risultato è il simbolo 0. Se tutti i simboli sono 1, allora il risultato è 1. In caso contrario la griglia viene ulteriormente suddivisa in *quarti* più piccoli.

Il metodo è applicato a tutte le sotto-griglie da sinistra in alto seguendo il senso orario. I 4 risultati di una stessa sotto-divisione vengono scritti uno dopo l'altro e sono delimitati da delle parentesi "...", come mostrato nell'immagine al centro e a destra. Il risultato finale è quindi la sequenza simboli che ne deriva.



Applica il metodo dei quarti al seguente immagine composta da  $8 \times 8$  pixel. Quale è il risultato finale?



- A) (1110)
- B) (11(1011)1)
- C) (111(1(1101)11))
- D) (111(1(1011)11))



## Soluzione

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

(111(1(1011)11))

L'applicazione del metodo dei quarti funziona così (v. immagine): innanzi tutto, i simboli nella griglia non sono tutti uguali, è necessario quindi suddividerla in 4 sotto-griglie quadrate. Il risultato finale deve quindi possedere questa forma:

$$( \langle Risultato 1 \rangle \langle Risultato 2 \rangle \langle Risultato 3 \rangle \langle Risultato 4 \rangle )$$

Nelle sotto-griglie 1, 2 e 3 tutti i simboli sono 1 e il loro risultato è dunque 1. Il risultato finale deve quindi possedere questa forma:

$$(111 \langle Risultato 4 \rangle )$$

La sotto-griglia 4 deve essere ulteriormente suddivisa. Nelle sotto-griglie 4.1, 4.3 e 4.4 tutti i simboli sono 1. Il risultato finale deve quindi avere questa forma:

$$(111(1 \langle Risultato 4.2 \rangle 11))$$

Infine la sotto-griglia 4.2 deve essere ancora suddivisa. Di conseguenza, dobbiamo considerare i singoli simboli binari il cui risultato è (1011). Dunque il risultato finale è:

$$(111(1(1011)11))$$

## Questa è l'informatica!

Più si utilizzano sistemi informatici quali PC, smartphone o tablet, più si creano quantità enormi di dati digitali. E anche se questi sistemi dispongono di capacità di memoria e di trasmissione dei dati sempre maggiori, non sono mai abbastanza. È dunque importante pensare a come comprimere testi, immagini, video, ecc. . . in modo da risparmiare più possibile la memoria, senza però perdere informazioni.

L'informatica applica numerosi metodi di compressione. Il metodo mostrato in questo compito funziona particolarmente bene se sono presenti grosse aree in cui i dati sono uguali. La suddivisione in quarti (e sotto-quarti) può essere compresa anche con l'aiuto dei cosiddetti quadtree. Un quadtree è una variante della struttura dati ad albero, che in informatica possiede una grossa importanza .

## Parole chiave e siti web

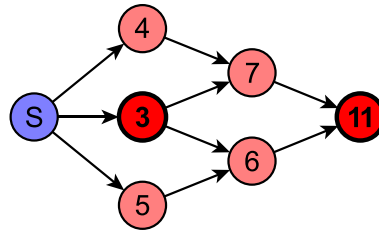
Quadtree, compressione di immagini

- <https://it.wikipedia.org/wiki/Quadtree>



## 38. Scorciatoia o strada più lunga?

L'immagine rappresenta la mappa delle strade a singola corsia di una città. In particolare, il numero rappresenta la lunghezza del percorso *minimo* che da S conduce fino a quell'incrocio.



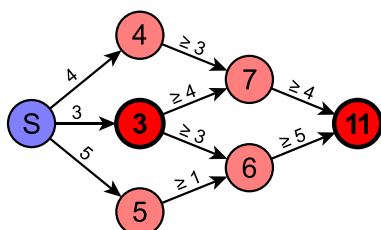
Quale delle seguenti affermazioni è vera per gli incroci segnati in rosso?

- A) La lunghezza del percorso minimo tra i due incroci è esattamente 8.
- B) La lunghezza del percorso minimo tra i due incroci è 8 o meno di 8.
- C) La lunghezza del percorso minimo tra i due incroci è 8 o più di 8.
- D) Non si può dire nulla di preciso sulla lunghezza del percorso minimo tra i due incroci.



## Soluzione

La risposta corretta è: “La lunghezza del percorso minimo tra i due incroci è 8 o più di 8.”  
Se la lunghezza del percorso minimo fosse meno di 8, allora la lunghezza tra S e l’incrocio segnato con “11” sarebbe meno di  $3 + 8 = 11$ . Il percorso minimo potrebbe però essere più grande di 8, poiché il cammino più corto tra S e l’incrocio segnato con “11” potrebbe passare da quelli segnati con “4” e “7” o quelli segnati con “5” e “6”. Dunque tutte le altre risposte sono sbagliate.



L’immagine qui sopra mostra la lunghezza minima di ogni tratto, considerati i valori dati.

## Questa è l’informatica!

A prima vista, si potrebbe pensare che il percorso minimo passa attraverso i due incroci considerati. Per esserne sicuri, bisognerebbe però conoscere la distanza effettiva tra loro, che purtroppo non abbiamo. Qui, infatti, viene indicato il percorso minimo da S e non la lunghezza del tratto. Per questo possiamo solo affermare che il percorso minimo tra i due incroci è maggiore o uguale a 8.

## Parole chiave e siti web

cammino minimo

- [https://it.wikipedia.org/wiki/Cammino\\_minimo](https://it.wikipedia.org/wiki/Cammino_minimo)



## 39. Display digitale

Un display digitale a 7 segmenti rappresenta le cifre nel modo seguente:



Ogni cifra è composta da un massimo di 7 segmenti. Supponiamo ora di avere un display che mostri una sola cifra e che la vista di esso sia parzialmente ostruita.

*Quali segmenti non possono essere nascosti affinché tutte le cifre possano sempre essere distinte?*





## Soluzione

La risposta corretta è:



Le coppie di cifre 1/7, 3/9, 5/6, 6/8 e 8/0 stanno alla base della soluzione di questo compito. Queste cifre, infatti, differiscono di un solo segmento, la cui vista, quindi, non deve essere ostruita, altrimenti non si potrebbe distinguere la coppia:



I due segmenti rimanenti non sono rilevanti per distinguere le cifre, dato che ogni coppia di cifre differisce anche per altri segmenti, oppure ha questi segmenti in comune.

## Questa è l'informatica!

Quando una parte di informazione può essere trascurata senza rendere il messaggio incompleto o incomprensibile, si parla di ridondanza. In generale, i testi ne sono un esempio concreto: piccoli cambiamenti non ne pregiudicano la leggibilità. In generale, la ridondanza incrementa lo spazio necessario per memorizzare un file, d'altro canto però rende le informazioni più affidabili contro modifiche involontarie, disturbi di trasmissione o errori diversi. La ridondanza, quindi, non è necessariamente negativa.

Con la "compressione dei dati" è generalmente possibile ridurre la ridondanza nei file. Questo consente di risparmiare spazio nel memorizzare e tempo nel trasferire dei dati. Di solito, però, un file compresso danneggiato è praticamente impossibile da recuperare. Anche nelle compressioni odierne, comunque, si aggiungono piccole ridondanze per permettere la correzione di errori minimi.

## Parole chiave e siti web

riconoscimento dell'informazione, compressione, ridondanza, cifre, display a 7 segmenti

- [https://it.wikipedia.org/wiki/Display\\_a\\_sette\\_segmenti](https://it.wikipedia.org/wiki/Display_a_sette_segmenti)
- [https://it.wikipedia.org/wiki/Ridondanza\\_\(ingegneria\)](https://it.wikipedia.org/wiki/Ridondanza_(ingegneria))





## 40. Suddivisione del codice

In un codice speciale pensato per i testi, ogni lettera viene sostituita da un numero composto dalle cifre da 0 a 9. Questo codice possiede una particolarità: ogni numero associato ad una lettera, non può iniziare con il numero completo assegnato ad un'altra lettera.

Per esempio, la lettera "X" viene codificata con "12", "Y" con "2". Siccome "12" non inizia con "2" e "2" non inizia con "12" questi codici sono validi. "Z" può essere codificata con "11", poiché gli altri due numeri già assegnati non iniziano con "11" e viceversa. "21" non sarebbe un codice valido per "Z", poiché "2", assegnato a "Y", corrisponde alla prima cifra di "21".

*La parola ANATRE è stata codificata con la sequenza di cifre 12112233321. Quale delle seguenti suddivisioni rappresentano ognuna una lettera della parola?*

- A) 12 11 22 33 32 1
- B) 1 21 1 22 33 321
- C) 1 21 12 2 33 321
- D) 1 21 1 22 3 3321
- E) 12 1 12 23 33 21



## Soluzione

La risposta corretta è:

*1 21 1 22 33 321*

Iniziamo dalle prime cifre a sinistra. Se “A” fosse codificata con “12”, “N” sarebbe associata necessariamente a “1” (poiché di seguito, avremmo un’altra “A” che dovrebbe essere codificata con “12”). La regola del prefisso sarebbe però contraddetta, poiché “12” inizia con “1”. È pure impossibile che “A” sia codificata con più di due cifre (ad esempio, “121”, “1211”, ...) poiché tale sequenza non è ripetuta una seconda volta, necessaria per la seconda “A”. Quindi “A” può essere codificata solo dal numero “1”.

“N” può allora essere codificata con “2”, “21” o “211223332”. “2” non è possibile, altrimenti la parola conterrebbe un’ulteriore “A” (ANAA...) e nemmeno “211223332”, altrimenti non ci sarebbero abbastanza numeri per completare la parola. Dunque “N” è necessariamente “21”.

Ora, la lettera “E” alla fine della parola non può essere codificata né da “1” né da “21” e quindi solo da “321”, “3321”, “33321”, “233321” o “2233321”. Gli ultimi due numeri possono essere scartati, in quanto non ne resterebbero abbastanza per completare la parola. Nemmeno “33321” è un codice valido, poiché “T” e “R” potrebbero solo associate entrambe al numero “2”. Se “E” fosse “3321”, “TR” verrebbe codificata con “223”. Ma “T” non potrebbe essere “2” (“21” è associato a “N”) e d’altro canto se fosse “22”, “R” non potrebbe essere “3” (poiché “E” sarebbe “3321”). Dunque “E” può solo essere “321”.

Rimane alla fine il numero “2233” per la codifica delle lettere “TR”. In analogia con quanto detto “T” deve essere “22” ed “R” “33”.

## Questa è l’informatica!

Il codice utilizzato in questo compito è un esempio di *codice prefisso*. Per “prefisso” si intende una sequenza di simboli (caratteri, numeri, lettere, ...) all’inizio di una sequenza più lunga. Nel codice prefisso, nessuna “parola” può iniziare esattamente con i simboli associati ad un’altra.

Nel codice prefisso le parole possiedono una lunghezza diversa. Grazie alla regola discussa nel compito, non sono necessari ulteriori simboli per fornire una suddivisione: è infatti sempre possibile capire quando una parola inizia e quando finisce.

Associando i codici più corti alle lettere maggiormente utilizzate, è possibile comprimere in modo efficiente un testo e dunque risparmiare spazio nella memoria.

Il codice di Huffman è un codice prefisso ottimale, usato in molti metodi di compressione quali JPEG, MP3 e ZIP.

## Parole chiave e siti web

codice prefisso, crittografia, crittoanalisi, “craccare” un codice

- [https://it.wikipedia.org/wiki/Codice\\_prefisso](https://it.wikipedia.org/wiki/Codice_prefisso)
- [https://it.wikipedia.org/wiki/Codifica\\_di\\_Huffman](https://it.wikipedia.org/wiki/Codifica_di_Huffman)



## A. Autori dei quesiti

 Andrea Adamoli	 Arnheiður Guðmundsdóttir	 J.P. Pretti
 Nursultan Akhmetov	 Yasemin Gülbahar	 Daniel Rakijašić
 Adil Aliyev	 Martin Guggisberg	 Chris Roffey
 Haim Averbuch	 Urs Hauser	 Frances Rosamond
 Khuyagbaatar Batsuren	 Hans-Werner Hein	 Kirsten Schlüter
 Wilfried Baumann	 Fredrik Heintz	 Victor Schmidt
 Bartosz Bieganski	 Juraj Hromkovič	 Eljakim Schrijvers
 Daphne Blokhuis	 Ungyeol Jung	 Masood Seddighin
 Eugenio Bravo	 Filiz Kalelioğlu	 Maiko Shimabuku
 Andrej Brodnik	 Dong Yoon Kim	 Taras Shpot
 Carmen Bruni	 Vaidotas Kinčius	 Martin Stangl
 Amaury A. Castro Jr.	 Jia-Ling Koh	 Seiichi Tani
 Marios Choudary	 Ivana Kosírová	 Gergely Tassy
 Anton Chukhnov	 Regula Lacher	 Monika Tomcsányiová
 Raluca Constantinescu	 Greg Lee	 Peter Tomcsányi
 Zsófia Csepregi-Horváth	 Milan Lukić	 Ahto Truu
 Valentina Dagienė	 Dario Malchiodi	 Willem van der Vegt
 Darija Dasović Rakijašić	 Hiroki Manabe	 Jiří Vaníček
 Christian Datzko	 Dimitris Mavrovouniotis	 Troy Vasiga
 Susanne Datzko	 Mattia Monga	 Nicolette Venn
 Janez Demšar	 Henry Ong	 Corina Elena Vint
 Olivier Ens	 Zsuzsa Pluhár	 Michael Weigend
 Hanspeter Erni	 Wolfgang Pohl	 Hongjin Yeh
 Michael Fellows	 Ilya Posov	 Momo Yokoyama
 Gerald Futschek	 Sergei Pozdniakov	 Khairul A. Mohamad Zaki



## B. Sponsoring: concorso 2017

**HASLERSTIFTUNG**

<http://www.haslerstiftung.ch/>

**ROBOROBO**

<http://www.roborobo.ch/>

**digitec.ch**

<http://www.digitec.ch/> & <http://www.galaxus.ch/>

**bischof  
berger**

<http://www.baerli-biber.ch/>

**verkehrshaus.ch**

<http://www.verkehrshaus.ch/>  
Museo Svizzero dei Trasporti

**Kanton Zürich  
Volkswirtschaftsdirektion  
Amt für Wirtschaft und Arbeit**

Standortförderung beim Amt für Wirtschaft und Arbeit  
Kanton Zürich

Information plus Automatik? Chunsch druus?  
Das ergibt Informatik.

i-factory (Museo Svizzero dei Trasporti, Lucerna)

**UBS**

<http://www.ubs.com/>  
Wealth Management IT and UBS Switzerland IT

**bbv**  
Software Services

<http://www.bbv.ch/>

**PRESENTEX**  
Das Geschenk - die gute Werbung

<http://www.presentex.ch/>



**PH LUZERN**  
**PÄDAGOGISCHE**  
**HOCHSCHULE**

<http://www.phlu.ch/>  
Pädagogische Hochschule Luzern

**ABZ**

AUSBILDUNGS- UND BERATUNGSZENTRUM  
FÜR INFORMATIKUNTERRICHT

<http://www.abz.inf.ethz.ch/>  
Ausbildungs- und Beratungszentrum für Informatikunterri-  
cht der ETH Zürich.

**n|w** Fachhochschule  
Nordwestschweiz

<https://www.fhnw.ch/de/die-fhnw/hochschulen/ph>  
Pädagogische Hochschule FHNW

**z** hdk  
Zürcher Hochschule der Künste  
Game Design

<https://www.zhdk.ch/>  
Zürcher Hochschule der Künste

  
ZUBLER & PARTNER AG  
I n f o r m a t i k

<http://www.zubler.ch/>  
Zubler & Partner AG Informatik

**senarclens**  
**leu+partner**  
strategische kommunikation

<http://senarclens.com/>  
Senarclens Leu & Partner



## C. Ulteriori offerte

010100110101011001001001  
0100000100101110101010011  
010100110100100101000101  
001011010101001101010011  
010010010100100100100001

**SSII**

[www.svia-ssie-ssii.ch](http://www.svia-ssie-ssii.ch)  
schweizerischervereinfürinformatikind  
erausbildung//sociétésuissepourl'infor  
matique dans l'enseignement//societàsviz  
zeraperl'informaticanell'insegnamento

Diventate membri della SSII <http://svia-ssie-ssii.ch/verein/mitgliedschaft/> sostenendo in questo modo il Castoro Informatico.

Chi insegna presso una scuola dell'obbligo, media superiore, professionale o universitaria in Svizzera può diventare membro ordinario della SSII.

Scuole, associazioni o altre organizzazioni possono essere ammesse come membro collettivo.