



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

**Quesiti e soluzioni 2017
5^o e 6^o anno scolastico**

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

A cura di:

Andrea Adamoli, Christian Datzko, Hanspeter Erni

010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001

SS!

www.svia-ssie-ssii.ch
schweizerischerverein für informatik und
erziehung // société suisse de l'inform
atique 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 18 novembre 2017 col sistema per la preparazione di testi L^AT_EX.



I quesiti sono distribuiti con Licenza Creative Commons Attribuzione – Non commerciale – Condividi allo stesso modo 4.0 Internazionale. Gli autori sono elencati a pagina 37.



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^o e 4^o 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^o e 4^o anno scolastico («Piccolo Castoro»)
- 5^o e 6^o anno scolastico
- 7^o e 8^o anno scolastico
- 9^o e 10^o anno scolastico
- 11^o al 13^o anno scolastico

Gli alunni iscritti al 3^o e 4^o 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

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

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



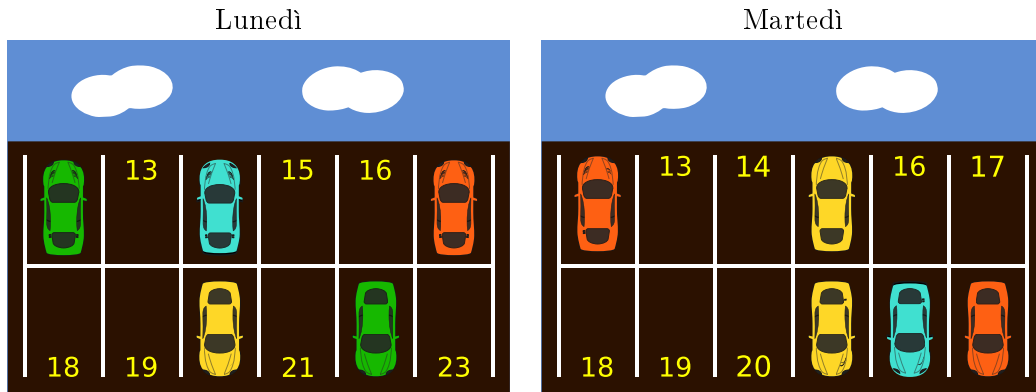
Indice

Hanno collaborato al Castoro Informatico 2017	i
Premessa	ii
1. Parcheggio	1
2. Cancelli binari	3
3. Servizio di messaggistica	5
4. Caccia alla fragola	7
5. Scambio di cani	9
6. Il castoro infortunato	11
7. 5 bastoncini	15
8. Esibizione di ballo	17
9. Il nome giapponese	21
10. Un piccolo programma	23
11. Gioielli	25
12. Honomakato	27
13. L'arte del bastone giapponese	31
14. La marmellata della nonna	33
15. La città delle rotonde	35
A. Autori dei quesiti	37
B. Sponsoring: concorso 2017	38
C. Ulteriori offerte	40



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

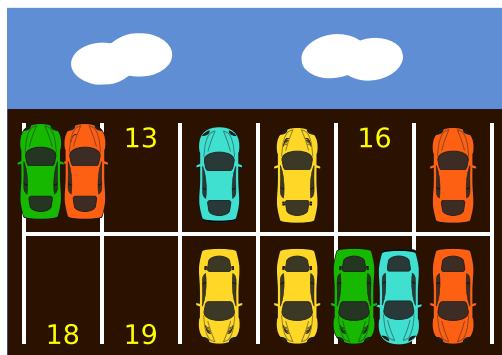


Soluzione

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.

Siti web e parole chiave

Bit, codice binario, OR logico, operatore logico

- https://it.wikipedia.org/wiki/Disgiunzione_logica



2. 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:



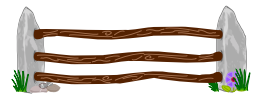
Siamo a casa, entra.



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.

Siti web e parole chiave

codifica, codice binario, standard

- https://it.wikipedia.org/wiki/Binary-coded_decimal
- https://it.wikipedia.org/wiki/Notazione_posizionale



3. 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 bit (2^{16} bit = 65'536 bit = 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.

Siti web e parole chiave

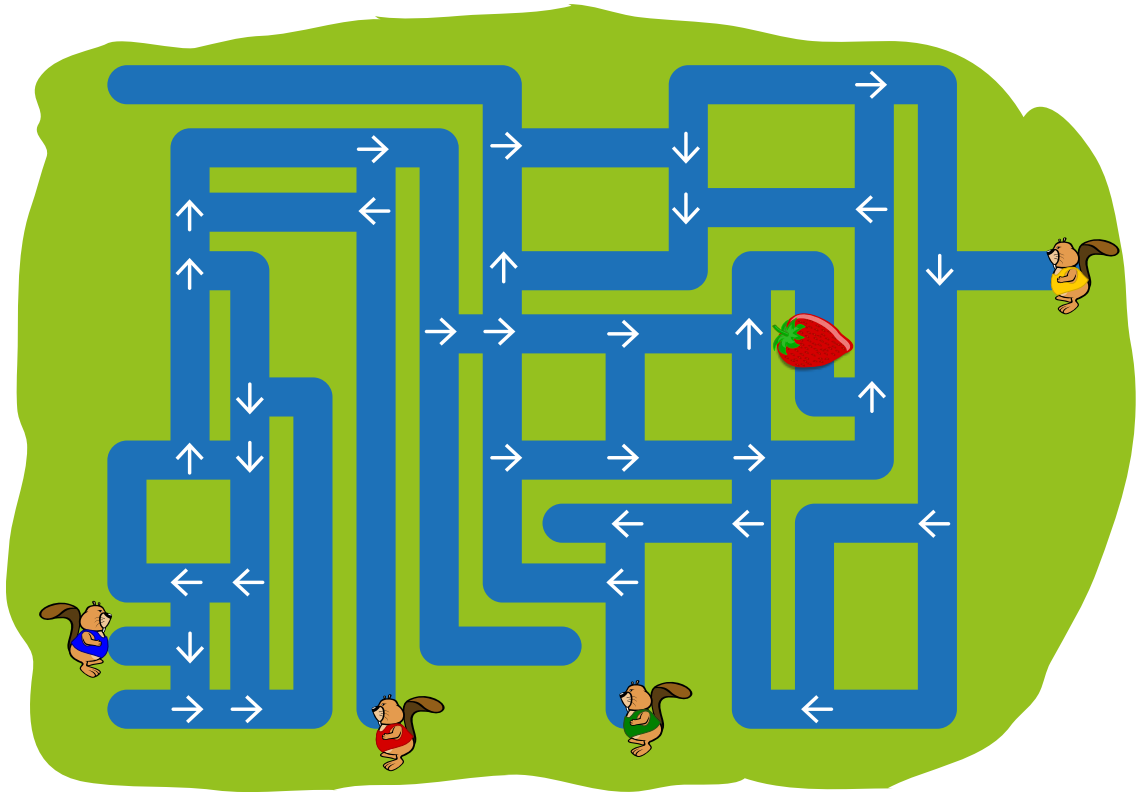
protocollo internet (IP), pacchetto di dati

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



4. Caccia alla fragola

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



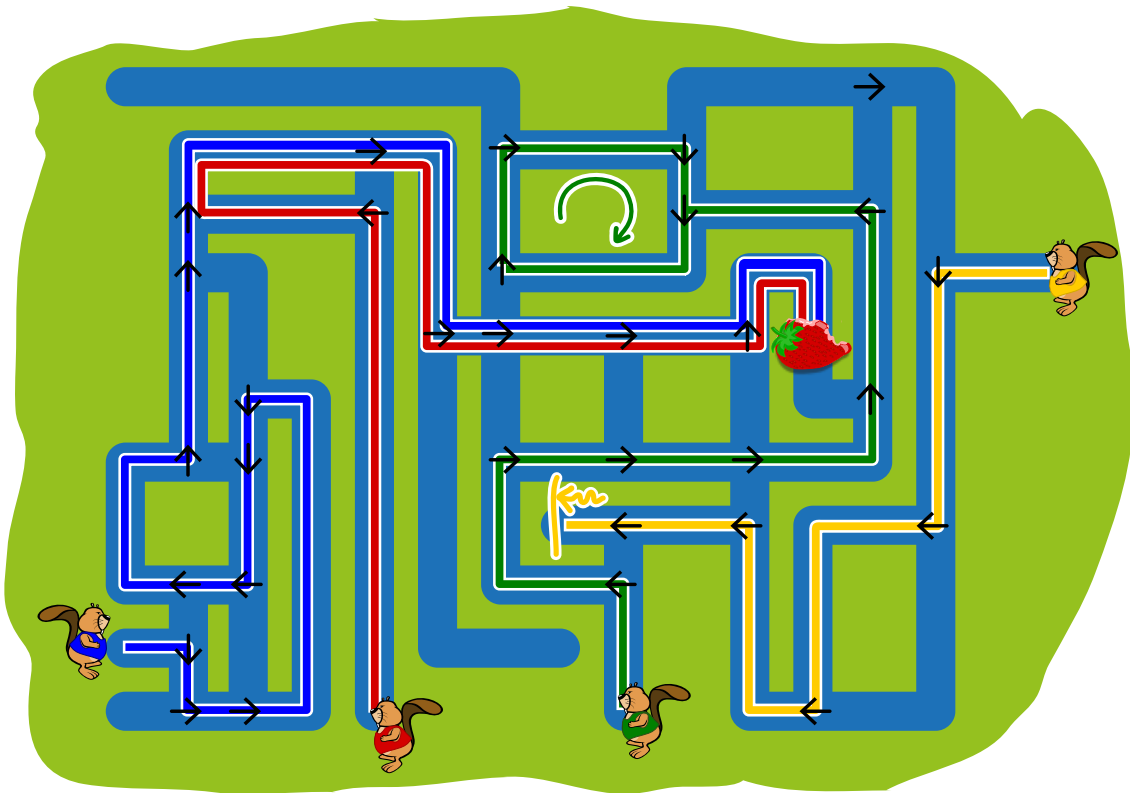
Quanti castori arriveranno fino alla fragola?

- 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.

Siti web e parole chiave

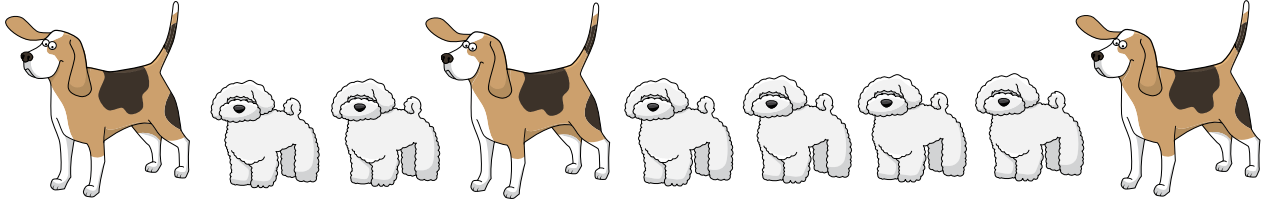
lettura dei grafi

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

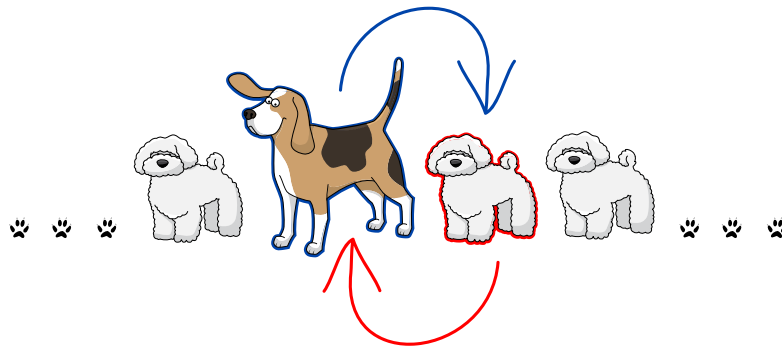


5. Scambio di cani

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".

Siti web e parole chiave

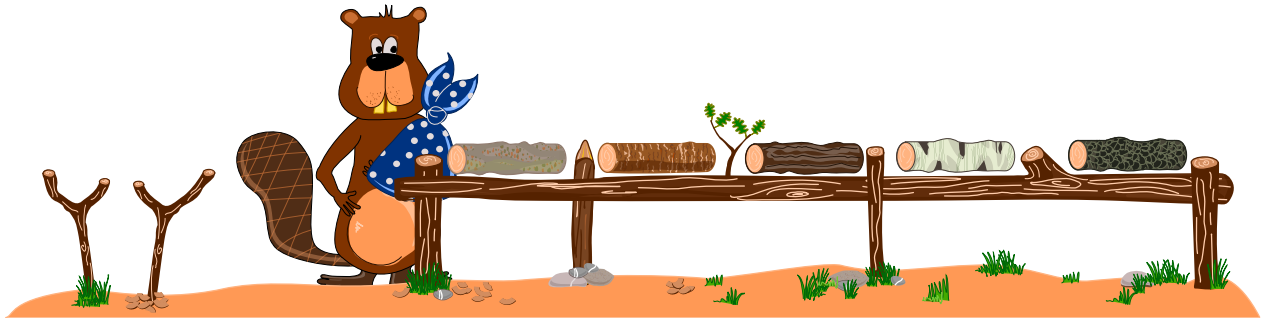
swap ("scambio")

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



6. 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.

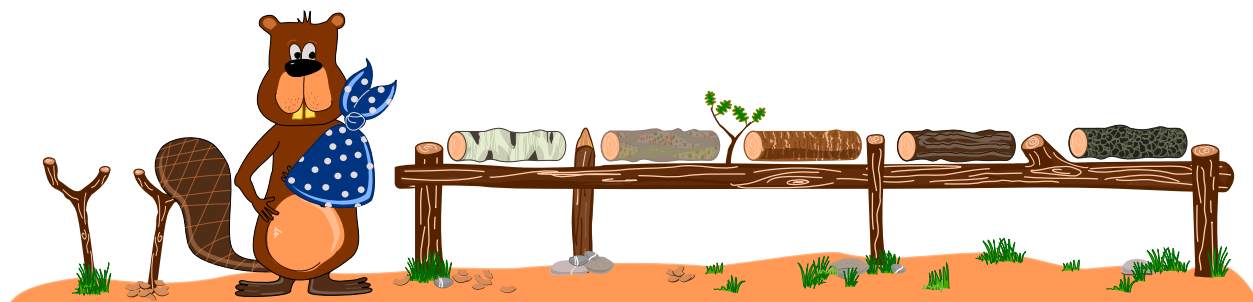


Aiuta David ad ordinare la propria collezione di ciocchi di legno dal più chiaro al più scuro, in modo che alla fine quello più chiaro sia a sinistra e quello più scuro a destra.



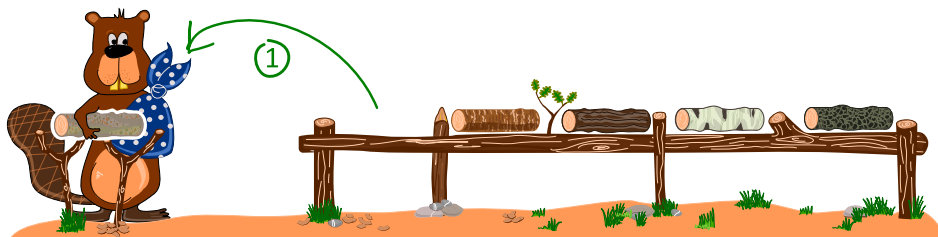
Soluzione

La risposta corretta è:

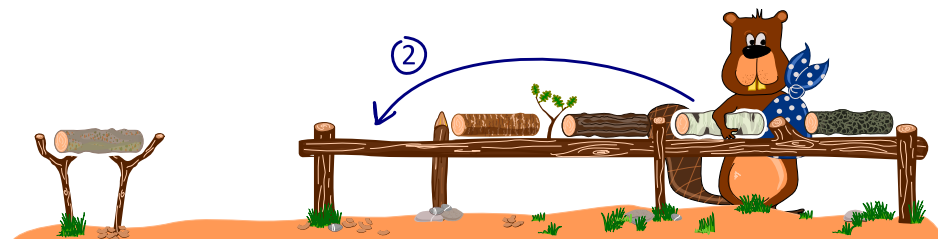


Per poter risolvere questo compito, David deve necessariamente utilizzare il sostegno a sinistra. Esistono molti metodi per ordinare i vari ciocchi, ma in ogni caso dobbiamo sempre scambiare di posto due ciocchi alla volta utilizzando il sostegno:

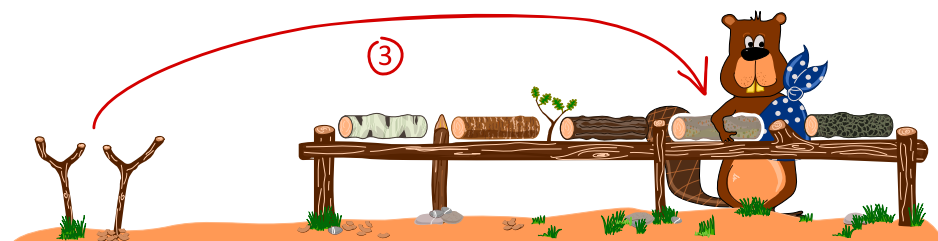
- Dapprima spostiamo un ciocco dalla sua posizione al sostegno.



- 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:

$$t \leftarrow a$$
$$a \leftarrow b$$
$$b \leftarrow t$$

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).

Siti web e parole chiave

Scambio di variabili, (algoritmi di) ordinamento

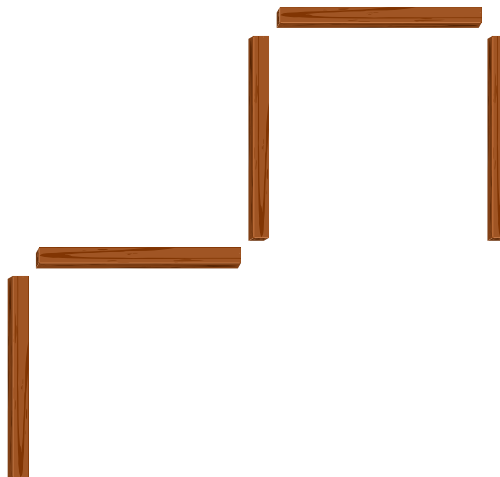
- [https://en.wikipedia.org/wiki/Swap_\(computer_programming\)](https://en.wikipedia.org/wiki/Swap_(computer_programming))
- https://it.wikipedia.org/wiki/Algoritmo_di_ordinamento



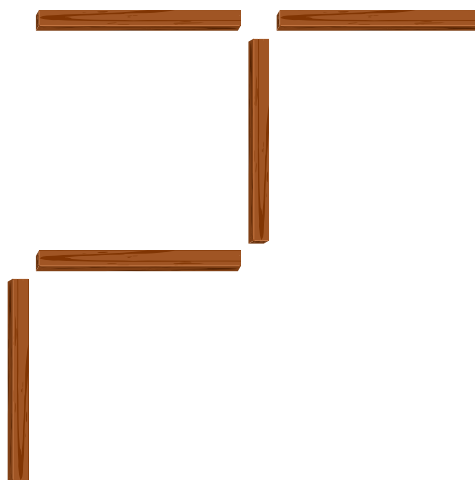


7. 5 bastoncini

5 bastoncini sono posti su un tavolo:

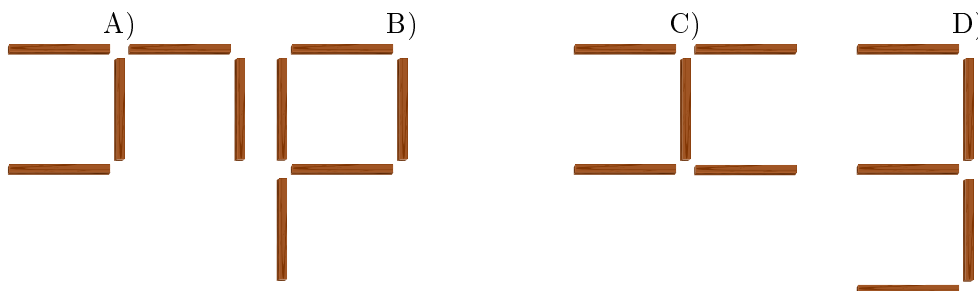


Nicole ne prende uno e lo mette in una nuova posizione. Adesso i bastoncini sono quindi collocati in questo modo:



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

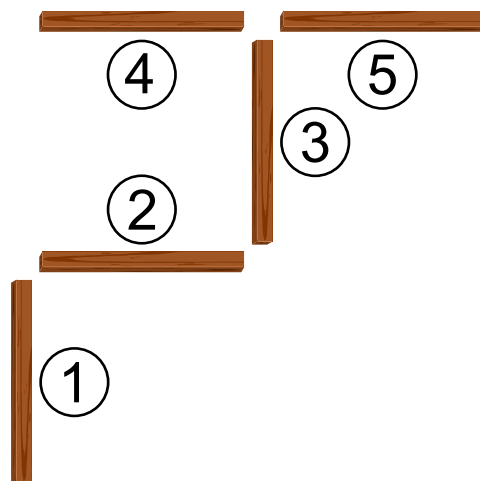
In quale delle seguenti posizioni non è possibile che siano collocati i bastoncini, adesso?





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.

Siti web e parole chiave

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))

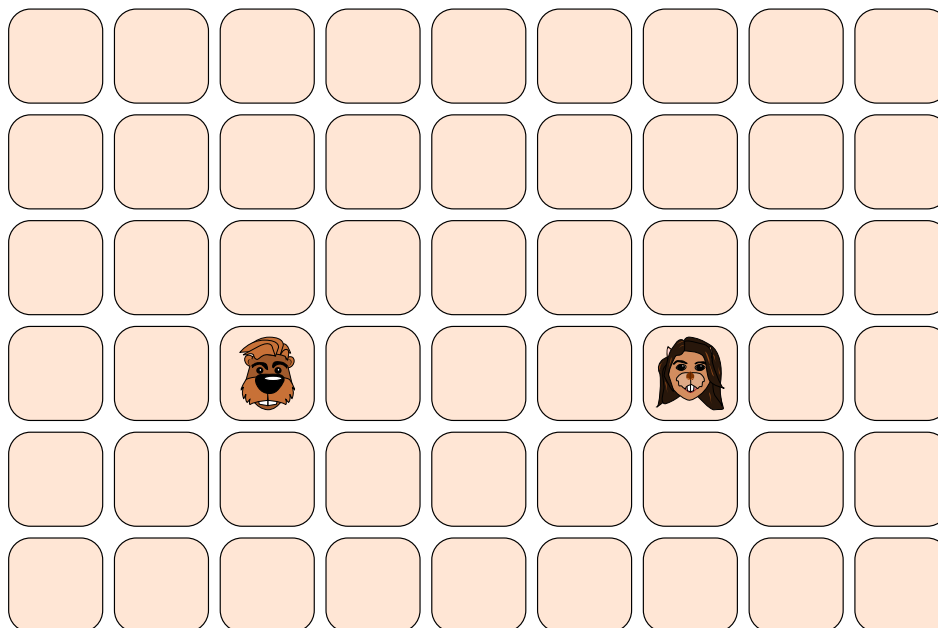


8. Esibizione di ballo

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:



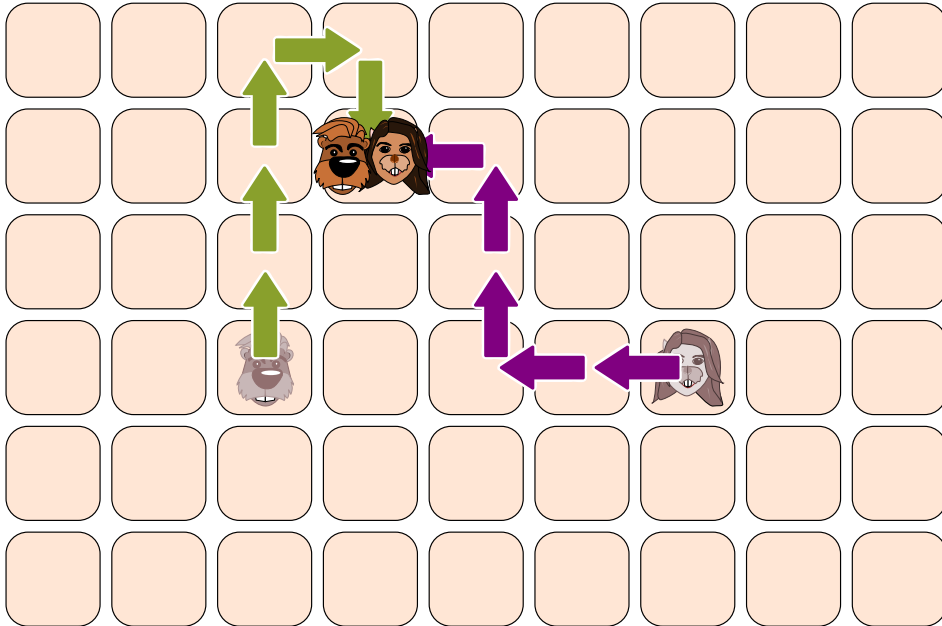
Quali delle seguenti reazioni del pubblico faranno in modo che lo scoiattolo e il castoro finiscano l'esibizione tutti e due nella stessa casella?

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



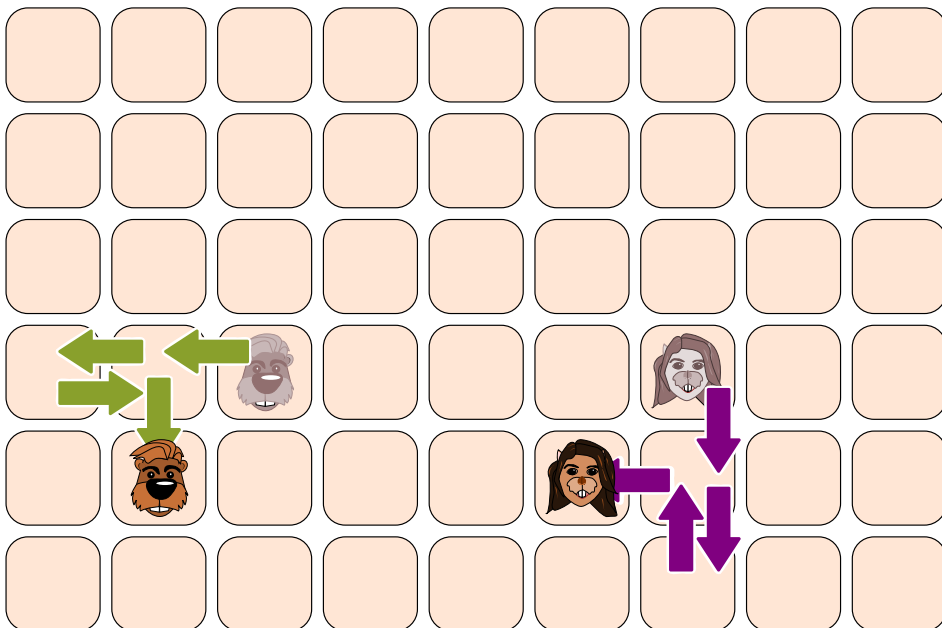
Soluzione

La risposta D) "Applausi! Grida!" è corretta:



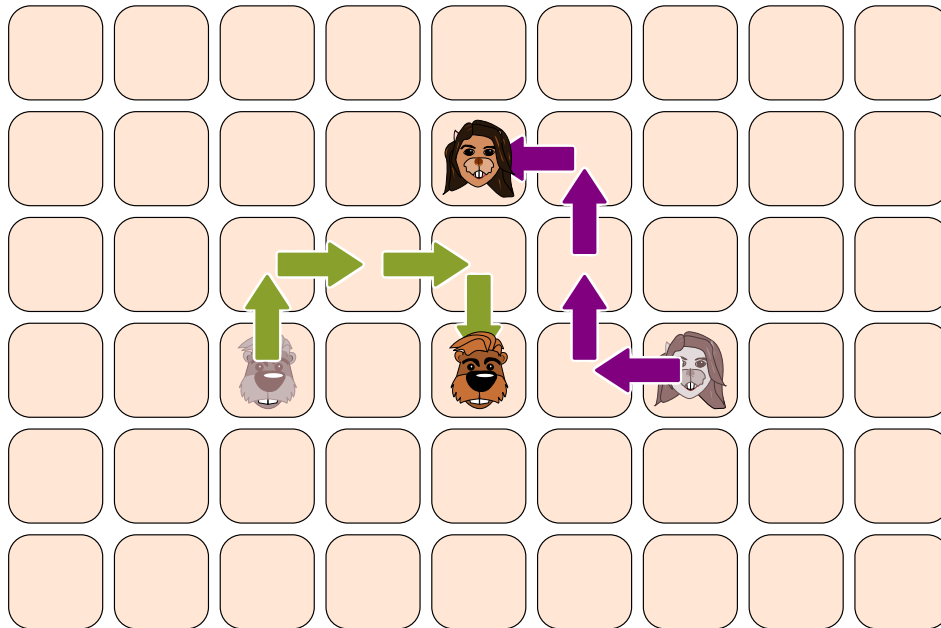
In tutti gli altri casi, lo scoiattolo e il castoro non terminano nella stessa casella:

A) "Fischi! Grida!":

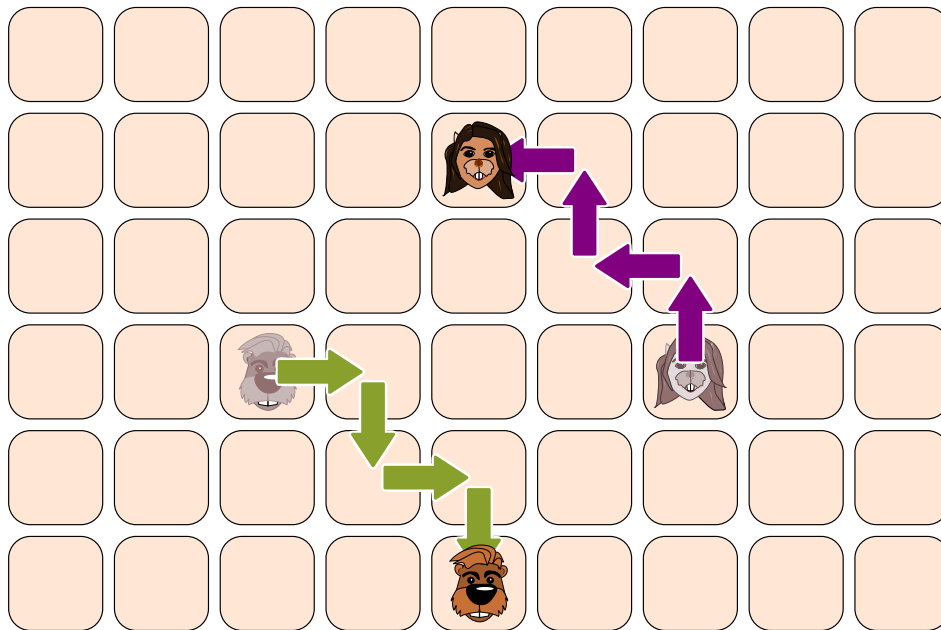




B) “Wow! Grida!”:



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.

Siti web e parole chiave

calcolo parallelo, semaforo, simulazioni

- 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>



9. Il nome giapponese

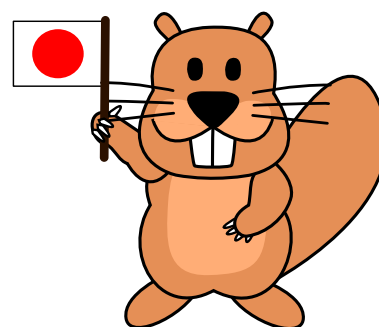
Un'amica giapponese ci racconta come, secondo una loro antica tradizione, dalle lettere del proprio nome si possa ottenere un nome spirituale, semplicemente sostituendo ogni lettera con delle sillabe associate:

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”.

Siti web e parole chiave

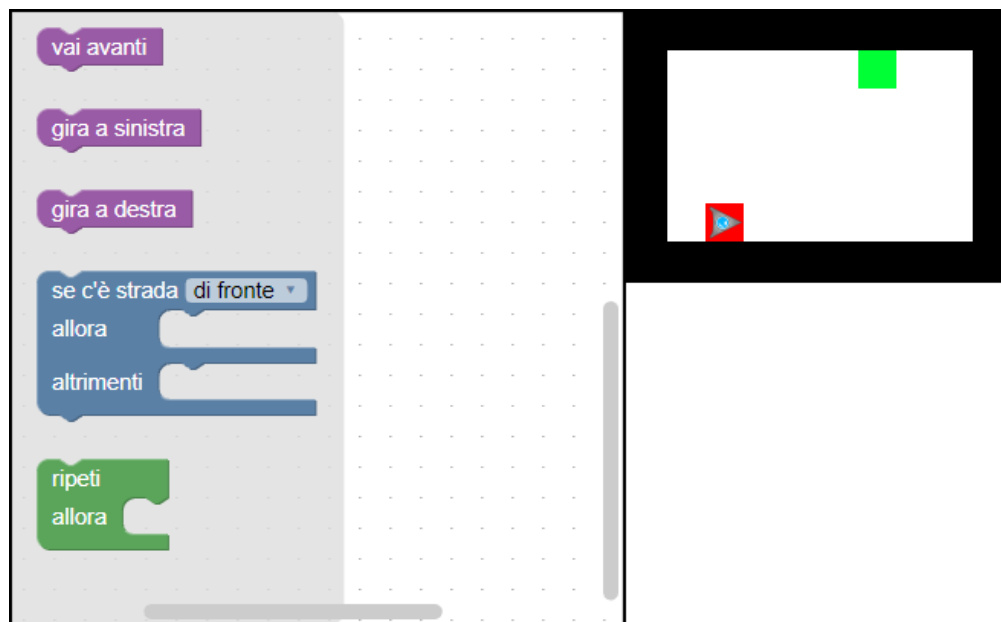
sistemi di riscrittura, semi-sistema di Thue, cifrario monoalfabetico

- https://en.wikipedia.org/wiki/Semi-Thue_system



10. 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.

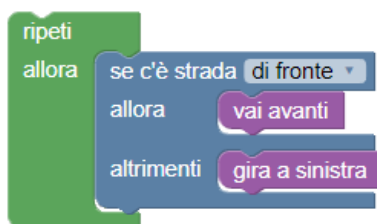


Trascina le istruzioni nel pannello di programmazione e uniscile nella giusta sequenza per permettere al robot di raggiungere il traguardo. Puoi usare solo un massimo di 4 istruzioni!



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 8×5 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.

Siti web e parole chiave

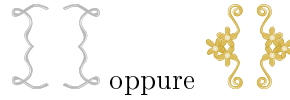
programmazione, cicli (loop), istruzioni condizionali, istruzioni

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

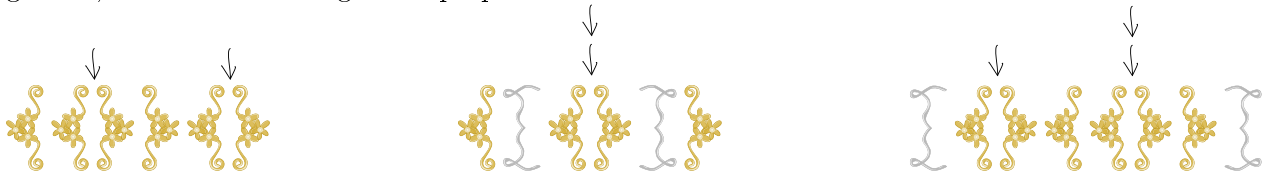


11. Gioielli

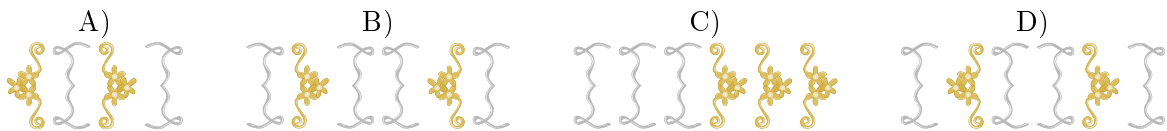
La famiglia Castoro produce artigianalmente dei gioielli in stile medioevale. Per fare questo, utilizza come base degli ornamenti a forma di parentesi, impiegati sempre a coppie. Per creare un gioiello, inizia da una delle seguenti coppie “di parentesi”:



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*".

Siti web e parole chiave

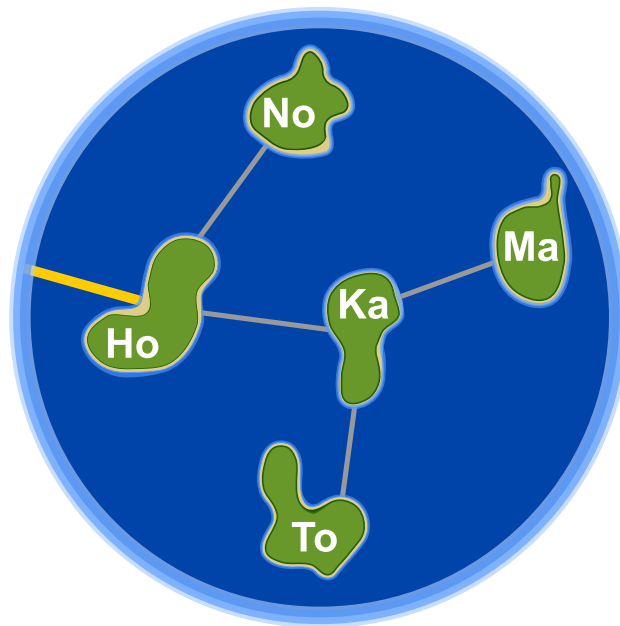
espressione "ben formata", sintassi, semantica

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



12. 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.

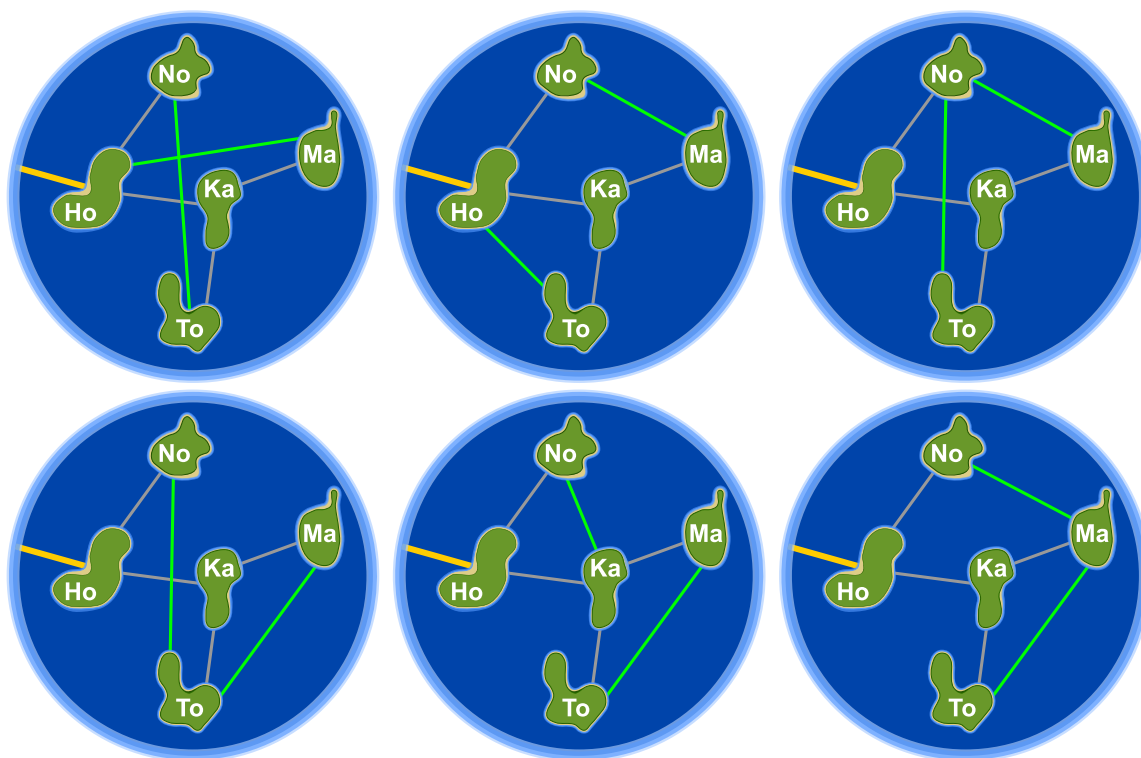


Soluzione

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.

Siti web e parole chiave

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

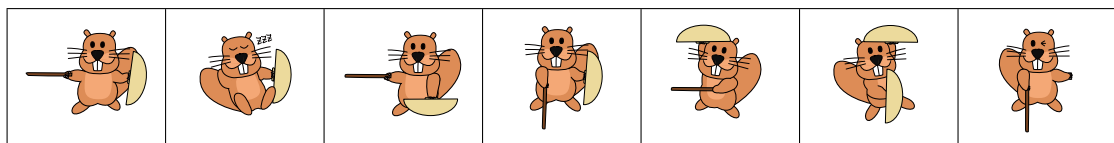
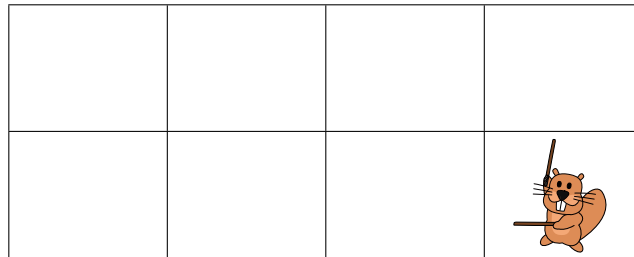
- https://en.wikipedia.org/wiki/Vertex_separator





13. 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.

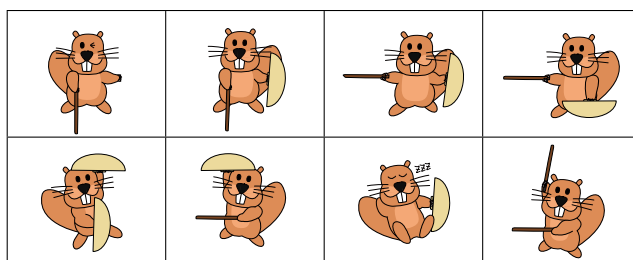


Associa le immagini degli amici di Lucia ai campi del cortile in modo che nella foto ogni bastone punti verso uno scudo.



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.

Siti web e parole chiave

pensiero logico, conclusioni logiche

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



14. 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 stare 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:

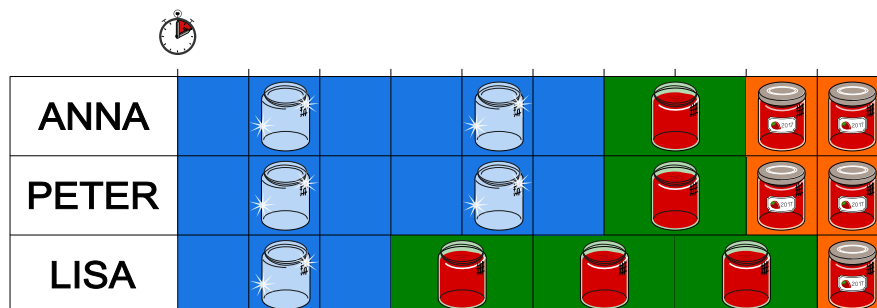


ANNA										
PETER										
LISA										



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.

Siti web e parole chiave

Scheduling (pianificazione), parallelismo (esecuzione parallela)

- https://en.wikipedia.org/wiki/Precedence_diagram_method
- <https://it.wikipedia.org/wiki/Scheduler>
- <http://www.cscs.ch/>

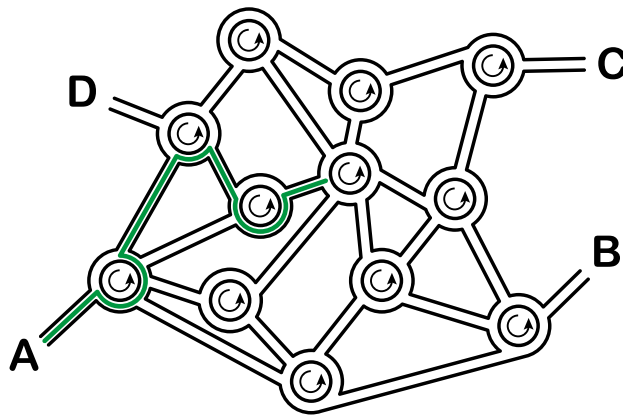


15. La città delle rotonde

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^a uscita.
- Alla rotonda successiva, prendi la 1^a uscita.
- A quella dopo, prendi la 2^a uscita.

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



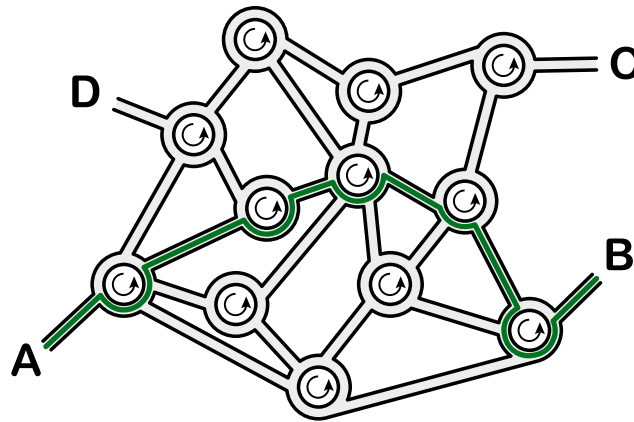
Dove conduce la spiegazione “3 1 3 2 3”, partendo dal punto A?

- 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.

Siti web e parole chiave

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

- https://it.wikipedia.org/wiki/Programmazione_imperativa



A. Autori dei quesiti

 Nursultan Akhmetov
 Adil Aliyev
 Haim Averbuch
 Khuyagbaatar Batsuren
 Wilfried Baumann
 Daphne Blokhuis
 Eugenio Bravo
 Andrej Brodnik
 Carmen Bruni
 Anton Chukhnov
 Darija Dasović Rakijašić
 Christian Datzko
 Susanne Datzko
 Janez Demšar

 Olivier Ens
 Hanspeter Erni
 Arnheiður Guðmundsdóttir
 Martin Guggisberg
 Hans-Werner Hein
 Fredrik Heintz
 Jia-Ling Koh
 Hiroki Manabe
 Wolfgang Pohl
 Sergei Pozdniakov
 J.P. Pretti
 Daniel Rakijašić
 Chris Roffey
 Frances Rosamond

 Kirsten Schlüter
 Eljakim Schrijvers
 Maiko Shimabuku
 Monika Tomcsányiová
 Peter Tomcsányi
 Ahto Truu
 Willem van der Vegt
 Troy Vasiga
 Michael Weigend
 Hongjin Yeh
 Momo Yokoyama
 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
schweizerischervereinfürinformatikind
erausbildung//sociétésuisse del'inform
atique 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.