



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

Quesiti e soluzioni 2016 11^o al 13^o anno scolastico

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

A cura di:

Andrea Adamoli, Christian Datzko, Hanspeter Erni

010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001

SS!I

www.svia-ssie-ssii.ch
schweizerischerverein für informatik
erziehung und ausbildung // société suisse de l'inform
atique dans l'enseignement // società sviz
zera per l'informatica nell'insegnamento



Hanno collaborato al Castoro Informatico 2016

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

Un particolare ringraziamento va a:

Juraj Hromkovič, Giovanni Serafini, Urs Hauser, Tobias Kohn, Ivana Kosírová, Serena Pedrocchi, Björn Steffen: ETHZ

Valentina Dagiene: Bebras.org

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

Anna Morpurgo, Violetta Lonati, Mattia Monga: Italia

Gerald Futschek: Austrian Computer Society, Austria

Zsuzsa Pluhár: ELTE Informatikai Kar, Ungheria

Eljakim Schrijvers, Daphne Blokhuis, Marissa Engels: 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 2016 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'01.11.2016. Questo quaderno è stato creato il 13 novembre 2016 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 39.



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 2016 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 2016	i
Premessa	ii
1. Il codice KIX	1
2. Filtro mediano	3
3. Stretta di mano	5
4. Gioco del Nim	7
5. Numeri di maglia ordinati	9
6. Cassy, la tartaruga	11
7. Montacarichi	15
8. Gioco con le biglie	19
9. Due possibilità	23
10. Incrocio di tram	25
11. La codifica delle bandiere	27
12. Lacci da scarpe	29
13. Pittura ricorsiva	31
14. Giocando con le forme	33
15. Scoiattoli egoisti	37
A. Autori dei quesiti	39
B. Sponsoring: concorso 2016	40
C. Ulteriori offerte	42



1. Il codice KIX

In Olanda i numeri di avviamento postale sono composti da 4 caratteri che possono essere lettere o cifre. Esiste in aggiunta anche un codice a barre associato, detto codice KIX. Ogni simbolo del codice KIX è suddiviso in una parte superiore (due barre lunghe e due corte) e una parte inferiore (di nuovo due barre lunghe e due corte). Per ogni carattere si uniscono queste due parti al centro. Nella tabella viene mostrato il codice KIX composto per 0, 7, G e Y:

	0	1	2	3	4	5
	6	7	8	9	A	B
	C	D	E	F	G	H
	I	J	K	L	M	N
	O	P	Q	R	S	T
	U	V	W	X	Y	Z

Il codice KIX per il numero di avviamento postale G7Y0 è quindi:

A quale numero di avviamento postale corrisponde questo codice KIX:



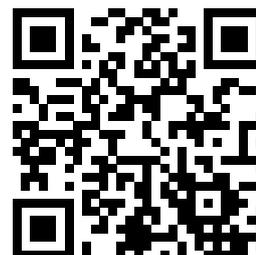
Soluzione

La risposta corretta è BC16:

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

Questa è l'informatica!

In Olanda si utilizza effettivamente il codice KIX. Codici facilmente leggibili dalle macchine permettono, ad esempio, di inviare pacchetti o lettere automaticamente. Codici simili sono spesso utilizzati: codici a barre (sui prodotti) per la scansione alla cassa, o codici QR, sviluppati dall'industria dell'auto per riconoscere le singole parti e oggi utilizzati come veicolo pubblicitario (esistono infatti molte applicazioni per smartphone che permettono di decifrarli). Cosa pubblicizza il codice QR posto qui a lato?



Siti web e parole chiave

codice KIX, codice a barre, codice QR

- <https://nl.wikipedia.org/wiki/KIX-code>
- https://it.wikipedia.org/wiki/Codice_a_barre
- <https://en.wikipedia.org/wiki/RM4SCC> (inglese)

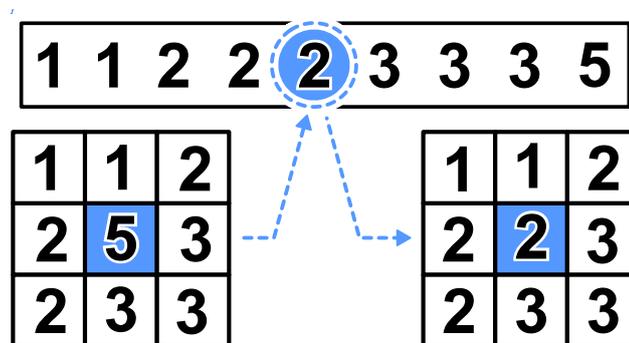


2. Filtro mediano

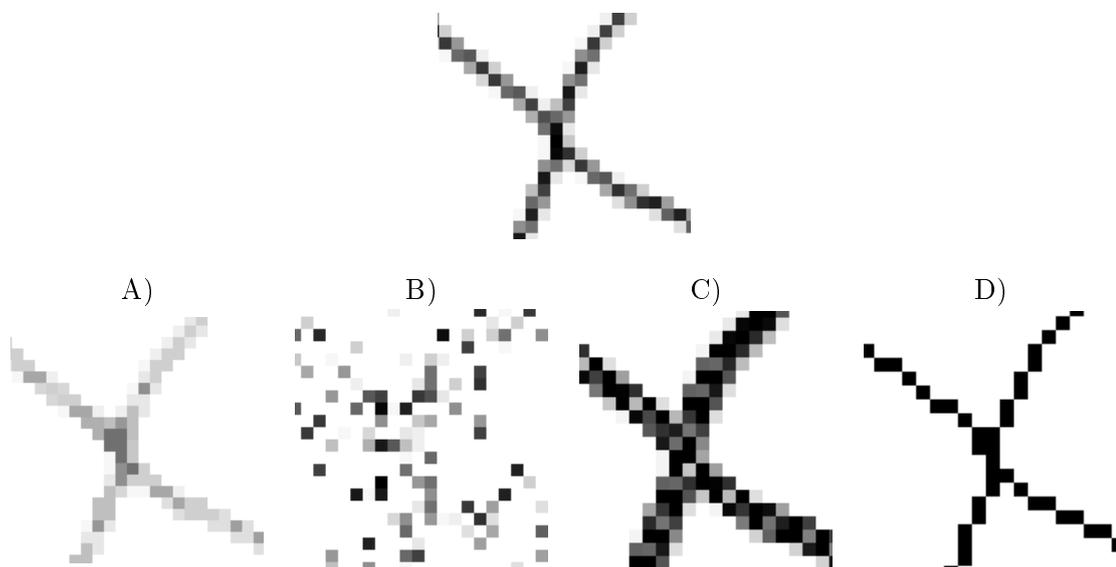
Un'immagine viene salvata come tabella indicante il tono di luminosità (da 1 a 5) per ogni pixel. Il valore 1 indica il nero, mentre il valore 5 indica il bianco. I valori tra 2 e 4 indicano toni di grigio sempre più chiari. Un cosiddetto "filtro mediano" modifica i toni di grigio di ogni pixel dell'immagine, in modo che...

- ...il valore del pixel e quello degli otto vicini vengono riportati in una sequenza ordinata...
- ...il pixel assume il valore del quinto elemento della sequenza (la mediana) quale nuovo tono di grigio.

In questo esempio, il pixel al centro passa dal valore 5 al valore 2:



Che apparirà questa immagine dopo aver applicato il filtro mediano?





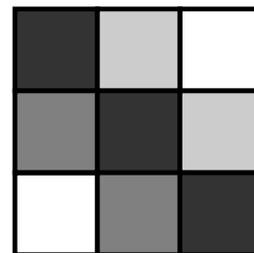
Soluzione

La risposta corretta è A):

Attraverso il filtro mediano vengono elaborati parti di immagine composte da nove pixel. I pixel neri al centro ottengono nuovi valori. Osservando l'esempio a destra è chiaro che solo 3 pixel possiedono una tonalità nera, la mediana è dunque necessariamente più chiara rispetto al pixel originale. Il nero sarà dunque convertito in un tono di grigio più chiaro.

Nell'immagine, in qualsiasi spezzone di 9 pixel, i "neri" sono sempre in minoranza, quindi l'immagine filtrata non potrà possedere toni di nero.

Questo si verifica solo nell'immagine A).



Questa è l'informatica!

Con l'elaborazione di immagini si vogliono applicare velocemente degli effetti visivi. Spesso si desidera ottenere immagini più vive o a fuoco. Altre volte si vogliono inserire effetti artistici per personalizzare le proprie foto. Tutti questi effetti si ottengono grazie a filtri dell'immagine.

Un esempio di tali filtri è quello mediano. Esso viene ad esempio impiegato per correggere errori nelle foto, dovuti magari a un piccolo difetto nei sensori di acquisizione. In seguito a questo effetto, l'immagine risulta più smussata e ogni pixel viene uniformato. Grazie a questo filtro possono essere dunque escluse determinate forme di "rumore".

Siti web e parole chiave

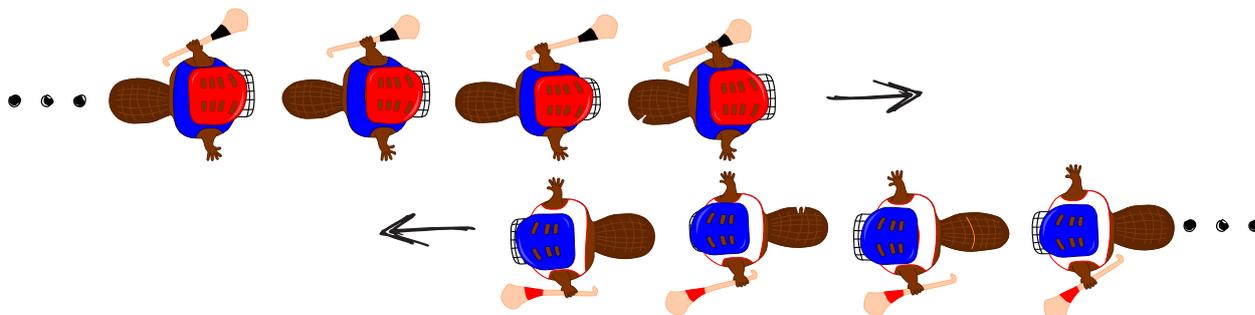
elaborazione dell'immagine, filtro mediano, livelli di grigio

- https://it.wikipedia.org/wiki/Filtro_mediano
- https://it.wikipedia.org/wiki/Livelli_di_grigio
- [https://it.wikipedia.org/wiki/Rumore_\(immagine\)](https://it.wikipedia.org/wiki/Rumore_(immagine))



3. Stretta di mano

I castori giocano volentieri all'Hurling, lo sport nazionale dell'Irlanda. Alla fine della partita i giocatori di ogni squadra si allineano in fila, uno dietro all'altro. Essi si dirigono poi verso la fila della squadra avversaria e si stringono la mano dicendo "Grazie per la partita!".



La stretta di mano funziona in questo modo: dapprima si stringono la mano i giocatori posti in cima alla propria fila, poi essi stringono la mano al giocatore che occupa il secondo posto della fila avversaria (come mostrato nella figura). Si continua in questo modo fino a quando anche gli ultimi giocatori della fila hanno stretto la mano a tutti quelli della fila avversaria.

Nell'Hurling ci sono 15 giocatori per ogni squadra. Per la stretta di mano tra due giocatori e lo spostamento verso il prossimo giocatore avversario si impiega 1 secondo.

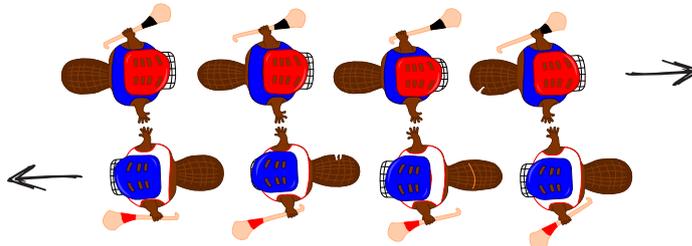
Quanti secondi si impiegano in totale per la stretta di mano tra tutti i membri delle due squadre?



Soluzione

La risposta corretta è 29.

La stretta di mano funziona così: il primo giocatore di ogni fila stringe la mano a tutti gli altri giocatori. Con 4 giocatori si impiegherebbero 4 secondi e si avrebbe la seguente situazione:



A quel punto, l'ultimo giocatore di ogni squadra avrebbe stretto la mano ad un giocatore avversario per la prima volta. Egli dovrebbe quindi stringere la mano ancora a tutti i restanti giocatori della fila avversaria, impiegando altri 3 secondi.

Generalizzando, possiamo quindi affermare che con n giocatori, impieghiamo dapprima n secondi e poi ancora $n - 1$ secondi. In totale abbiamo dunque $n + n - 1 = 2n - 1$ secondi. Per 15 giocatori si impiegano allora $15 + 15 - 1 = 29$ secondi per stringersi la mano.

Questa è l'informatica!

Per le squadre dell'Hurling composte da 15 giocatori, possiamo calcolare esattamente quanto tempo impiega complessivamente la stretta di mano. Quanto impiegherebbero, invece, due squadre di Hockey con 22 giocatori ciascuno? Possiamo ancora utilizzare l'algoritmo impiegato per l'Hurling? Sarebbe davvero utile poter disporre di un algoritmo in grado di valutare il tempo di esecuzione in generale, senza dover esaminare in dettaglio ogni caso.

L'informatica ricerca intensamente algoritmi generali per calcolare il tempo di esecuzione. Le analisi effettuate forniscono delle espressioni matematiche contenenti una variabile n associata alla grandezza dei dati. Per il nostro esempio, otteniamo questa espressione sostituendo il "numero di giocatori" (15) con n : $2n - 1$. In questo modo possiamo facilmente calcolare il tempo di esecuzione anche per squadre con più o meno giocatori. Per 22 giocatori, impiegheremmo $2n - 1 = 2 \cdot 22 - 1 = 43$ secondi, per 40 giocatori 79 secondi e così via.

L'espressione $2n - 1$ è una funzione lineare. L'algoritmo della "stretta di mano" appartiene quindi alla classe degli algoritmi con tempi di esecuzione lineari, designati con $O(n)$. Come sarebbe, però, se la stretta di mano avvenisse in modo diverso? Per esempio, se i giocatori si stringessero la mano singolarmente (e non in contemporanea). Ebbene, questo algoritmo apparterebbe alla classe $O(n^2)$ e le squadre di Hurling impiegherebbero $15^2 = 225$ secondi (ca. 4 minuti). Se l'algoritmo avesse poi un tempo di esecuzione esponenziale, ovvero se fosse della classe $O(2^n)$ si impiegherebbero $2^{15} = 32768$ secondi, ovvero ca. 9 ore. In questo caso gli spettatori sarebbero tornati a casa da molto tempo. E quindi utile, in questi casi, domandarsi se non sia possibile svolgere determinate operazioni in parallelo per risparmiare tempo.

Siti web e parole chiave

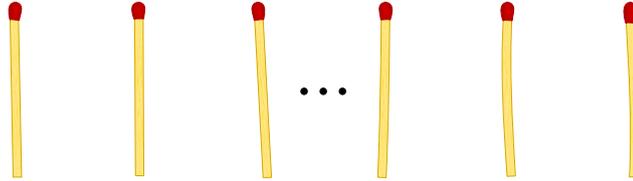
complessità temporale, analisi del tempo di esecuzione (run-time)

- <https://it.wikipedia.org/wiki/Run-time>
- https://it.wikipedia.org/wiki/Complessità_temporale



4. Gioco del Nim

Beat e il suo amico giocano al Nim: 13 fiammiferi sono posti inizialmente su un tavolo e alternativamente i due giocatori ne tolgono 1, 2 o 3. Il giocatore che prende l'ultimo fiammifero è il vincitore.



Aiuto: Se sul tavolo ci sono ancora 4 fiammiferi, il giocatore di turno non può più vincere. Questa situazione deve quindi essere evitata.

Beat inizia, quanti fiammiferi deve togliere per poter vincere il gioco?

- A) 1
- B) 2
- C) 3
- D) Non ha importanza.



Soluzione

La risposta corretta è 1. Sul tavolo restano quindi 12 fiammiferi. L'amico di Beat prende 1, 2 o 3 fiammiferi e al prossimo turno Beat dovrà fare in modo di lasciarne 8 sul tavolo. Di nuovo, l'amico prenderà 1, 2 o 3 fiammiferi e Beat ne lascerà sul tavolo 4. A questo punto Beat ha vinto.

Se Beat dovesse prendere 2 o 3 fiammiferi, il suo amico potrebbe fare in modo di lasciarne 4 sul tavolo, analogamente a quanto descritto sopra. A quel punto Beat non potrebbe più vincere.

Questa è l'informatica!

Nella teoria dei giochi, il gioco del Nim o il famoso gioco del "dilemma del prigioniero" sono utilizzati come modelli per analizzare problemi strategici del mondo reale e trovare delle soluzioni. Nell'economia di mercato, ad esempio, le conoscenze in questo campo servono ad ottimizzare i prezzi di vendita. Sebbene un prezzo più basso possa facilitare le vendite, spesso diminuisce il guadagno per ogni articolo. Al contrario, un prezzo alto aumenta il guadagno per ogni articolo, ma può anche far diminuire le vendite e quindi il profitto totale. Con i modelli della teoria dei giochi si possono simulare le possibili reazioni degli acquirenti al cambiamento dei prezzi. L'importanza della teoria dei giochi nell'economia è testimoniata dai numerosi premi Nobel attribuiti a ricerche in questo campo.

Siti web e parole chiave

gioco del Nim, teoria dei giochi, albero di decisione

- <https://it.wikipedia.org/wiki/Nim>
- <http://it.goobix.com/giochi-online/nim/> (versione alternativa a quella proposta nel quesito)
- https://it.wikipedia.org/wiki/Dilemma_del_prigioniero



5. Numeri di maglia ordinati

Le figure inserite qui sotto mostrano due squadre, ognuna delle quali composta da 15 giocatori. Entrambe le squadre indossano maglie con i numeri stampati sopra. I giocatori della prima squadra sono ordinati per numero, mentre i giocatori della seconda no.

Squadra 1:



Squadra 2:



Come si può verificare nella maniera più rapida quali numeri sono presenti sia nella squadra 1 che nella squadra 2?

- A) Passo in sequenza i numeri della squadra 1. Per ognuno di essi verifico se nella squadra 2 esiste lo stesso numero.
- B) Passo in sequenza i numeri della squadra 2. Per ognuno di essi verifico se nella squadra 1 esiste lo stesso numero.
- C) È uguale da quale squadra iniziare: entrambi i metodi sono ugualmente veloci.
- D) Dapprima bisogna sapere quali numeri non sono in comune.. Sottraggo questo numero a 15 e ottengo il numero cercato.



Soluzione

La risposta corretta è B). In una serie ordinata si può individuare un certo valore più velocemente che in una serie non ordinata. Un computer utilizzerebbe solo $\log_2(n)$ passaggi di ricerca invece di n . Quindi la risposta B) necessita nel caso peggiore $n \cdot \log_2(n)$ passaggi.

Nella risposta A) non si potrebbe scorrere la lista di numeri della squadra 1 velocemente, in quanto dovremmo occuparci di ogni numero individualmente. Quindi la risposta A) necessita di n^2 passaggi nel caso peggiore, molto più di $n \cdot \log_2(n)$ del caso B).

In virtù di quanto scritto sopra, la risposta C) è errata.

La risposta D) non fornisce di norma la risposta cercata e dunque deve essere scartata.

Questa è l'informatica!

In una serie ordinata si possono trovare dei valori più velocemente che in una serie non ordinata. Come cerchi, ad esempio, la maglia numero 9 nella squadra 1? Cerchi il giocatore centrale che ha il numero 17. 9 è più piccolo, quindi guardi a sinistra. Di nuovo, cerchi il giocatore centrale della parte sinistra della squadra, il quale possiede il numero 7. Allora guardi a destra e trovi velocemente il numero 9. Il trucco consiste nel dimezzare ogni volta lo spazio di ricerca, ciò che in informatica è detto "ricerca binaria". Dato che è possibile utilizzare la ricerca binaria solo sulle collezioni di dati ordinate, gli "algoritmi di ordinamento" sono molto importanti.

Nella spiegazione data sopra n si riferisce al numero di dati da ricercare (nel nostro caso 15).

Siti web e parole chiave

ordinamento, ricerca binaria (o dicotomica)

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



6. Cassy, la tartaruga

La tartaruga Cassy vive nelle terre di Reticolandia, in un campo formato da una griglia di 5 celle x 5 celle. Per vivere, si ciba volentieri di insalata. Ogni giorno sul campo crescono nuove piante di insalata, ma Cassy non conosce in quale punto esatto. Per trovarle, la mattina Cassy inizia la ricerca dal centro del campo e segue determinate istruzioni elencate in un blocco.

Fai in modo che Cassy visiti tutte le celle del campo. Scegli le istruzioni sulla sinistra cliccandogli sopra. Puoi utilizzare le istruzioni più di una volta.

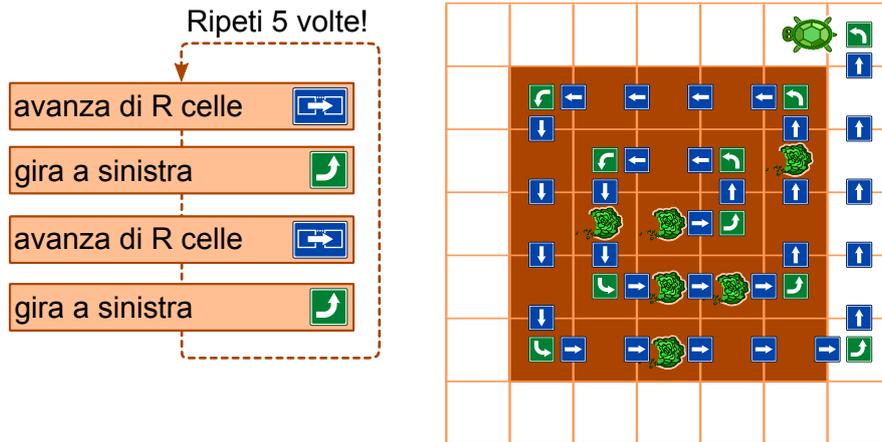
- R è una variabile numerica. Quando viene eseguita la prima volta nel blocco di istruzioni, R ha il valore 1, la seconda volta il valore 2, e così via.
- Cassy può lasciare il campo, ma non può uscire da Reticolandia.
- Cliccando su “Verifica”, puoi verificare il tuo programma.



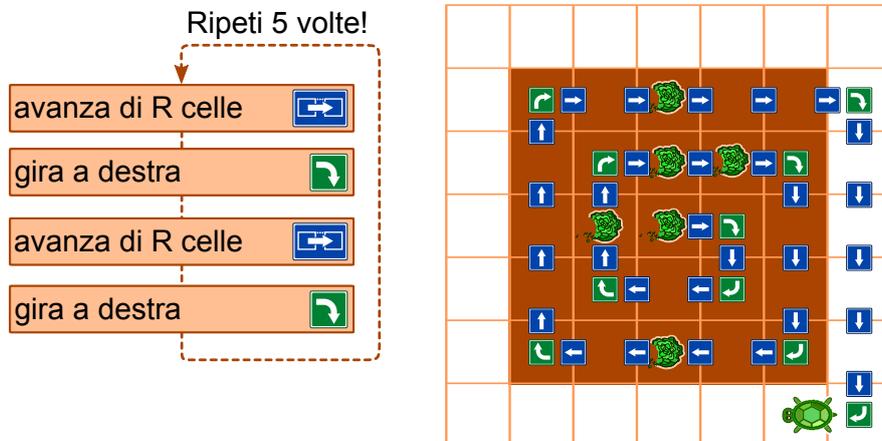
Soluzione

È possibile inserire nel blocco al massimo 4 istruzioni che vengono ripetute 5 volte. Questo significa che il cammino della tartaruga deve seguire una quello di una spirale. Esistono 4 diverse sequenze di istruzioni che Cassy potrebbe seguire per visitare tutte le celle del campo:

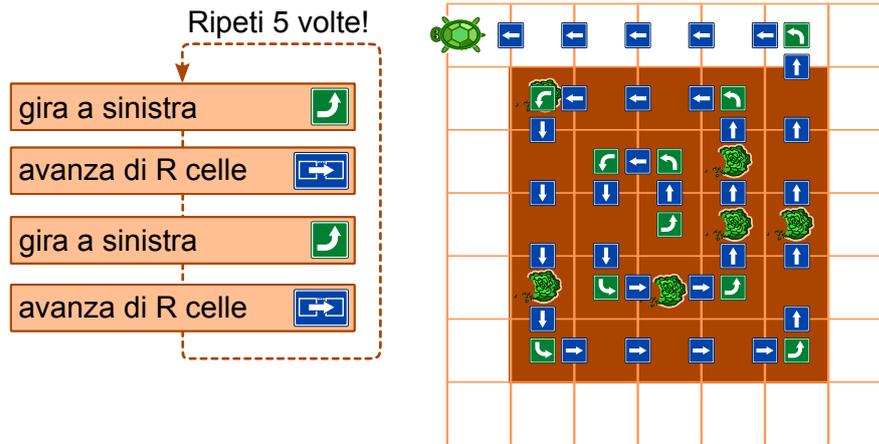
- avanza di R celle, gira a sinistra, avanza di R celle, gira a sinistra



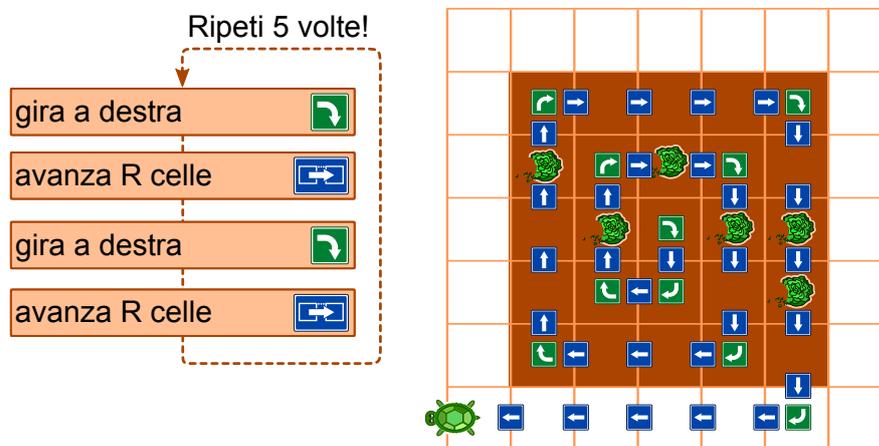
- avanza di R celle, gira a destra, avanza di R celle, gira a destra



- gira a sinistra, avanza di R celle, gira a sinistra, avanza di R celle



- gira a destra, avanza di R celle, gira a destra, avanza di R celle



Questa è l'informatica!

Per questo compito deve essere scritto un vero e proprio programma per computer. Esso è formato da una *sequenza di quattro singole istruzioni*, ripetute 5 volte. Queste ripetizioni sono dette *cicli (o loop)*. Nel nostro caso si tratta di un *count-controlled loop* (ovvero un ciclo controllato da un contatore), dato che R varia da 1 a 5. Un computer esegue queste istruzioni in sequenza. Tutti i linguaggi di programmazione supportano i cicli, oltre ad altre istruzioni che possono controllarne l'esecuzione, come ad esempio la chiamata a sotto-programmi (procedure).

Se il programma è corretto, il computer esegue esattamente ciò che desideri. In caso contrario, il computer eseguirà sempre le istruzioni, ma la tartaruga non si comporterà come ti aspetti. Normalmente, infatti, un computer non è in grado di capire se il risultato di una programmazione è quello voluto oppure no.

Siti web e parole chiave

Turtle-Graphics

- <http://www.turtlegrafik.ch/> (solo inglese o tedesco)
- <http://primalogo.ch/> (solo tedesco o francese)



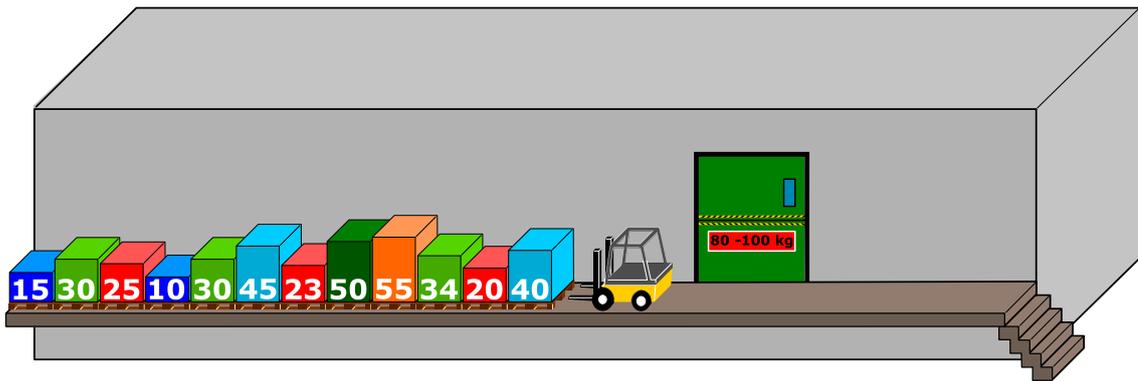
- <http://coesi.inf.usi.ch/eventi-trascorsi> (“Introduzione all’informatica e alla programmazione”)
- <https://xlogo.tuxfamily.org/it/index-it.html>



7. Montacarichi

Su una rampa di carico sono posizionati dei bancali (palette) affiancati. Essi, da sinistra a destra, pesano rispettivamente 15 kg, 30 kg, 25 kg, 10 kg, 30 kg, 45 kg, 23 kg, 50 kg, 55 kg, 34 kg, 20 kg, 40 kg. La rampa è talmente stretta che solo il bancale in cima alla fila può essere preso e spostato dal muletto.

I bancali vengono caricati su un montacarichi che li trasporta fino al magazzino. Il montacarichi entra in funzione non appena il carico raggiunge gli 80 kg. Esso, però, non può sollevare più di 100 kg in totale. Una volta scaricati i bacali, il montacarichi torna alla rampa vuoto.



Come detto, sul montacarichi viene caricato il bancale più vicino nella fila iniziale. Se il peso totale supera i 100 kg, il muletto prende l'ultimo bancale caricato e lo appoggia sull'altro lato della rampa. Altrimenti il bancale rimane sul montacarichi.

Quando tutti i bancali di sinistra sono stati portati al magazzino, il muletto comincia spostare i bancali infilati precedentemente sulla destra, in modo analogo a quanto fatto con la fila di sinistra.

Quale affermazione è corretta tra quelle esposte qui di seguito?

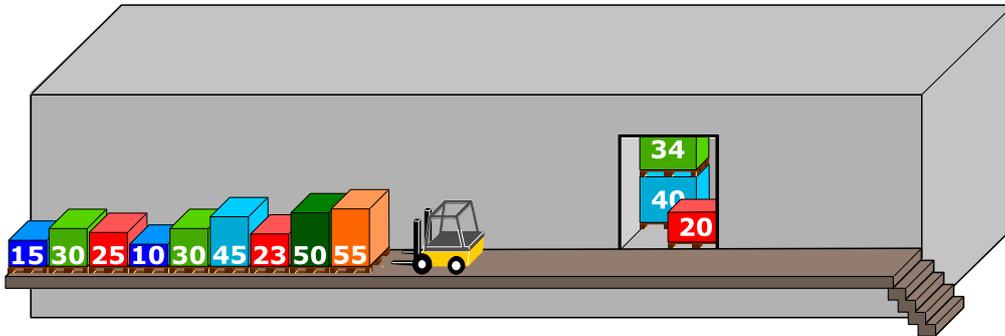
- A) Al secondo viaggio del montacarichi il peso complessivo ammontava a 98 kg.
- B) Non si è mai formata una fila a destra del montacarichi.
- C) In un viaggio del montacarichi il peso complessivo ammontava a 100 kg.
- D) Il montacarichi ha fatto in totale 5 viaggi.
- E) Non è possibile, con la procedura descritta sopra, spostare tutti i bancali nel magazzino.



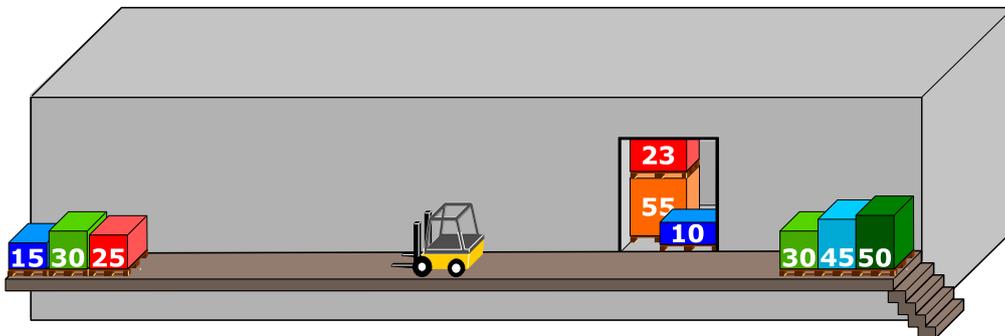
Soluzione

La risposta corretta è C):

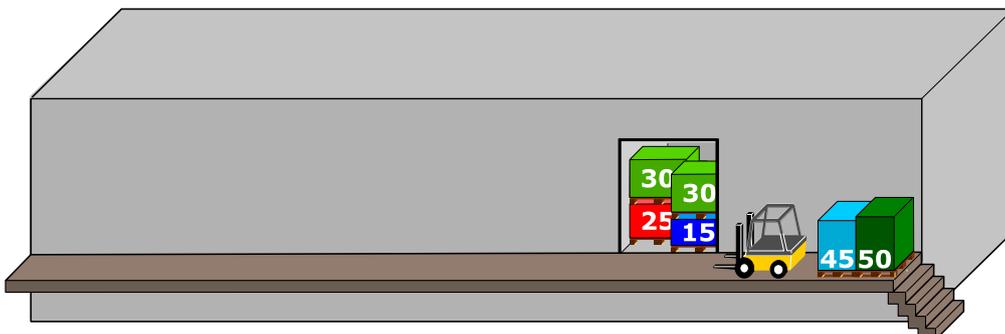
- Il primo carico è composto dai primi tre bancali. Questi pesano: $40\text{ kg} + 20\text{ kg} + 34\text{ kg} = 94\text{ kg}$.



- Il secondo carico è composto inizialmente da un bancale (55 kg). Quando viene caricato il successivo (50 kg), il peso complessivo eccede i 100 kg. Il muletto, quindi, lo toglie mettendolo sulla destra. Il prossimo bancale della fila di sinistra (23 kg) viene caricato e il peso totale raggiunge: $55\text{ kg} + 23\text{ kg} = 78\text{ kg}$. Il bancale successivo (45 kg) risulta di nuovo essere troppo pesante e dunque è spostato sulla fila di destra. La stessa operazione viene compiuta con il bancale che segue (30 kg). Infine, con bancale da 10 kg, il peso totale supera gli 80 kg ($55\text{ kg} + 23\text{ kg} + 10\text{ kg} = 88\text{ kg}$) e il carico può essere trasportato al magazzino.

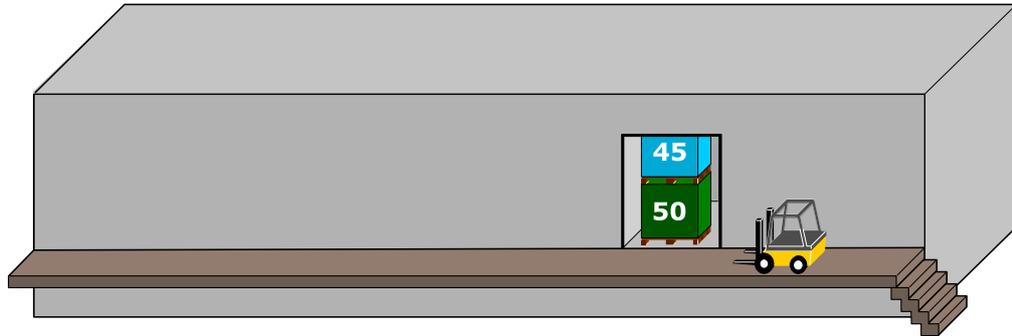


- Gli ultimi 3 bancali ($25\text{ kg} + 30\text{ kg} + 15\text{ kg} = 70\text{ kg}$) vengono caricati sul montacarichi. Dato che il peso è inferiore agli 80 kg, viene aggiunto anche il primo bancale della fila di destra (30 kg), raggiungendo così un totale di 100 kg.





- A questo punto vengono caricati gli ultimi due bancali, per un peso totale di: $45\text{ kg} + 50\text{ kg} = 95\text{ kg}$. Questo è il quarto e ultimo viaggio del montacarichi.



In virtù di questa spiegazione, le risposte A), B), D) ed E) sono chiaramente errate.

Questa è l'informatica!

Siccome la rampa è stretta, solo il primo bancale della fila può essere caricato. Questo comportamento corrisponde a quello di una pila. Quando si impilano in successione degli oggetti solo quello in cima può essere prelevato e, analogamente, solo in cima possono essere aggiunti altri oggetti. Anche in informatica simili strutture di dati vengono dette “pile” (o, più spesso, si usa il termine inglese “stack”). Nel nostro caso vengono formate tre pile: quella di sinistra (con primo elemento tutto a destra), quella di destra (primo elemento a sinistra) e quella nel montacarichi (primo elemento quello più in alto).

Le pile sono abbastanza comuni anche nella vita reale: quando da bambini si costruiscono delle torri con i Lego, solo il mattoncino in cima può essere rimosso senza danni e solo in cima possono venire aggiunti altri mattoncini; anche un cono gelato può essere considerato una pila, dato che l'ultima pallina aggiunta è la prima ad essere mangiata; un ulteriore esempio è la cronologia salvata durante la navigazione sul web: quando si clicca “Indietro” si torna all'ultima pagina visitata.

Il principio alla base di questo comportamento è detto LIFO “last in – first out” (l'ultimo a entrare è il primo a uscire). Esso si contrappone al FIFO “first in – first out” (il primo a entrare è anche il primo a uscire), un principio utilizzato nelle code d'attesa (per esempio dal dottore, anche se per fortuna i casi urgenti hanno la priorità).

Siti web e parole chiave

struttura dei dati, pila (o stack), algoritmo, LIFO, FIFO

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





8. Gioco con le biglie

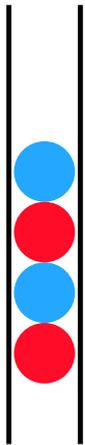
Emil gioca a un nuovo solitario sul suo computer. Il gioco inizia con un minimo di 3 biglie colorate (rosse o blu) impilate in un tubo.

Dopo aver premuto un tasto, le due biglie più in basso cadono fuori dal tubo. Allo stesso tempo, delle nuove biglie vengono impilate dall'alto. A dipendenza del colore della biglia posta inizialmente più in basso, possono accadere due cose:

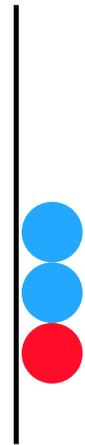
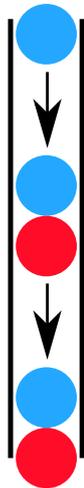
Se la biglia più in basso era rossa, una nuova biglia blu viene impilata in alto:

Se la biglia più in basso era blu, vengono impilate 3 nuove biglie rispettivamente di colore rosso, blu e rosso:

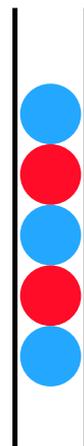
prima



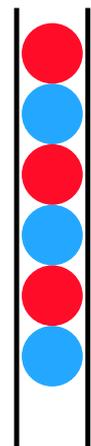
dopo



prima



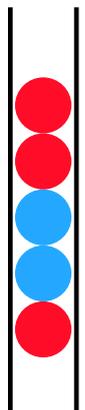
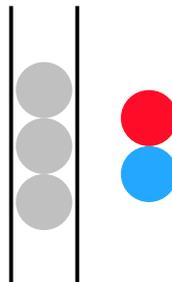
dopo



Fino a quando ci sono almeno 3 biglie nel tubo, Emil continua a premere il tasto. Il gioco termina non appena ci sono meno di 3 biglie nel tubo.

Se inizialmente nel tubo dovessero esserci le biglie mostrate nella figura a destra, dopo cinque pressioni del tasto resterebbero solo 2 biglie blu e il gioco terminerebbe.

Trascina nel tubo le biglie con colori opportuni per fare in modo che il gioco non termini mai.

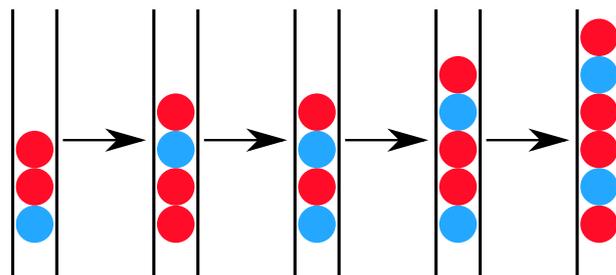
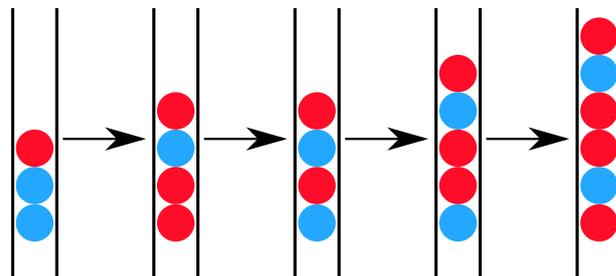
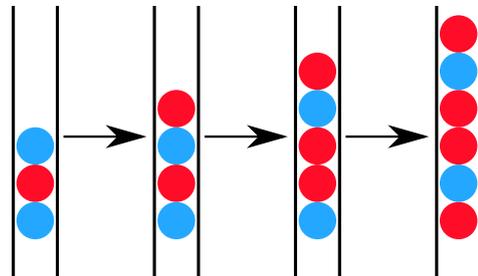
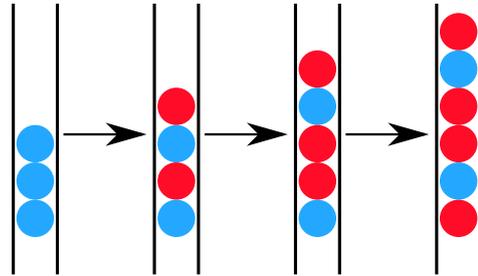




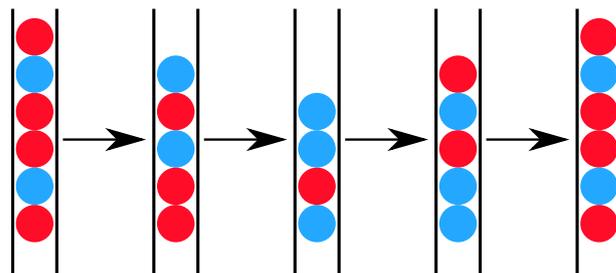
Soluzione

Se la biglia posta più in basso nella pila di 3 biglie fosse rossa, il gioco terminerebbe già dopo la prima pressione del tasto, poiché resterebbero solo 2 biglie.

Al contrario, se l'ultima biglia fosse blu il gioco continuerebbe all'infinito. Con al massimo quattro pressioni del tasto, si otterrebbe la una pila di 6 biglie con colori, risp., rosso-blu-rosso-rosso-blu-rosso:



Con questa configurazione si entrerebbe poi in un ciclo infinito:





Questa è l'informatica!

Il gioco presentato in questo compito fu preso ad esempio da Emil Leon Post per dimostrare come il processo di riscrittura di catene di simboli (stringhe) possa condurre a cicli infiniti (processi senza termine). Emil Leon Post (1897-1954) era un matematico e studioso della logica di origine polacca, che pubblicò numerosi contributi scientifici nel campo dell'informatica teorica e, più precisamente, della logica proposizionale.

Un modello di riscrittura, come quello del nostro esempio, è descritto attraverso una grammatica formale. In essa vengono definite le regole di sostituzione. Nel nostro caso, la grammatica era definita da 4 regole (dove X è una qualsiasi biglia di colore rosso o blu):

$$Xbb \rightarrow rbrX$$

$$Xrb \rightarrow rbrX$$

$$Xbr \rightarrow bX$$

$$Xrr \rightarrow bX$$

Siti web e parole chiave

modelli computazionali, linguaggi formali, regole di produzione, stringhe, sistemi di elaborazione

- http://esolangs.org/wiki/Post_canonical_system (inglese)
- https://en.wikipedia.org/wiki/Tag_system (inglese)
- https://en.wikipedia.org/wiki/Post_canonical_system (inglese)



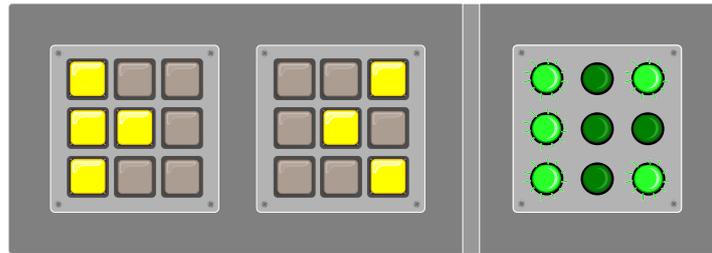


9. Due possibilità

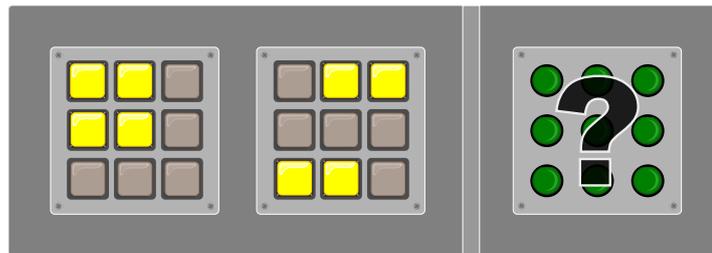
Erich ha trovato un vecchio apparecchio elettronico. Sulla parte sinistra ci sono 2 campi, ognuno dei quali ha 9 tasti. Sulla parte destra c'è invece un campo con 9 luci LED. Premendo i tasti dei campi a sinistra si accendono o si spengono i LED sulla destra.

Erich osserva che la posizione del LED che si accende o spegne è correlata alla stessa posizione e alla combinazione dei tasti dei due campi di sinistra.

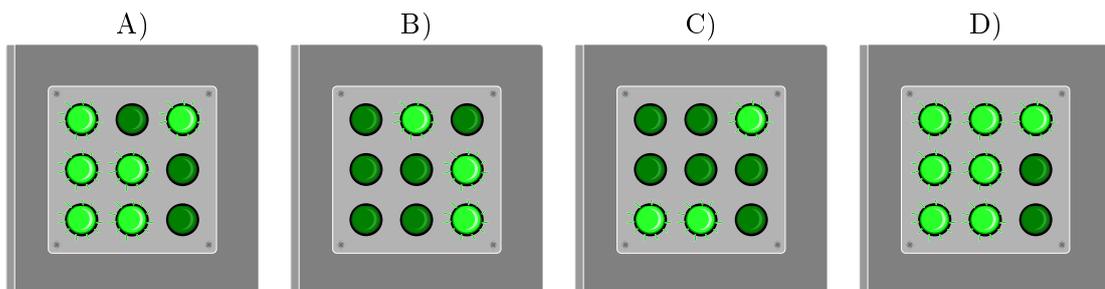
Al momento la combinazione di tasti (sinistra) e i LED accesi (destra) è la seguente:



Erich cambia quindi la combinazione come segue:



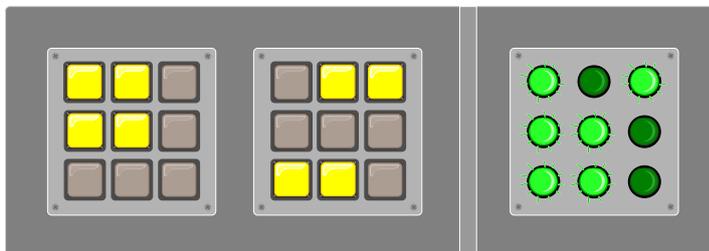
Quali LED si accenderanno sulla destra?





Soluzione

La risposta corretta è A). La regola è: quando i tasti nella stessa posizione nei due campi a sinistra sono entrambi accesi o spenti, allora il LED nella posizione corrispondente a destra è spento. Altrimenti è acceso.



Questa è l'informatica!

In tutti i campi dell'informatica si utilizza spesso la logica binaria per valutare la correttezza o l'inesattezza di determinate asserzioni in una certa circostanza. Questo, naturalmente, con il presupposto che ci siano solo due possibilità di risposta, una terza possibilità è esclusa. Le funzioni logiche, infatti, possono fornire solo due valori: “vero” o “falso”.

Nel nostro compito abbiamo usato la funzione logica XOR (dall'inglese eXclusive OR, “oppure esclusivo”). Essa funziona in questo modo: quando due asserzioni sono entrambe “vere” o entrambe “false” (risp. “accese” o “spente”) allora il risultato è “falso” (“spento”). Negli altri casi (ovvero “falso-vero”, “vero-falso”) essa restituisce un valore “vero”. Nel compito bisognava applicare questa funzione 9 volte (i tasti) e esprimerne il risultato a destra (LED).

Siti web e parole chiave

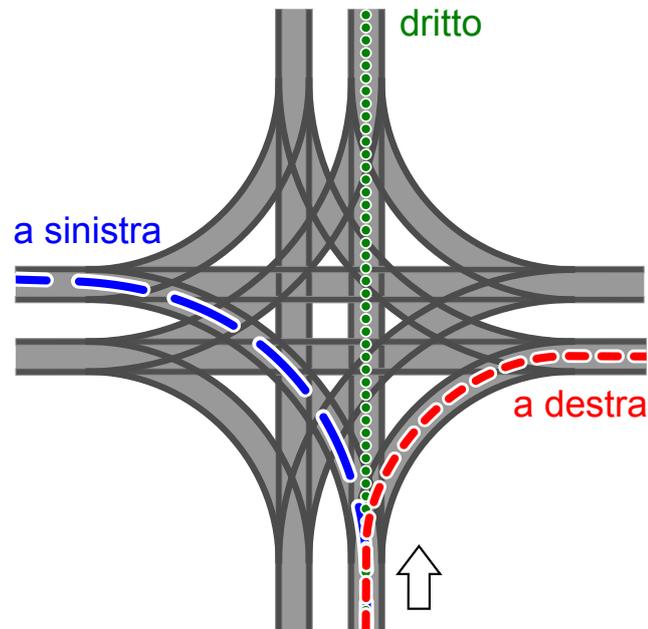
logica binaria, funzione XOR

- https://it.wikipedia.org/wiki/Algebra_di_Boole
- https://it.wikipedia.org/wiki/Disgiunzione_esclusiva
- https://en.wikipedia.org/wiki/XOR_gate (inglese)



10. Incrocio di tram

Sulle strade di St. Pietroburgo circolano dei tram. In città c'è un incrocio che permette di imboccare qualsiasi strada: i tram possono arrivarci da tutte le direzioni e svoltare a destra, a sinistra o continuare dritti.



La direzione in cui vanno i tram è determinata dalla posizione di alcuni scambi. Tale posizione è descritta dalle parole DIRITTO, SINISTRA e DESTRA. Per esempio: la combinazione SINISTRA-DIRITTO-SINISTRA-DESTRA significa che gli scambi sono posti in modo che un tram vada a sinistra, il prossimo (in senso orario) dritto, il successivo (sempre in senso orario) a sinistra e il quarto tram a destra.

Può anche succedere che all'incrocio arrivino tram da tutte le direzioni allo stesso momento.

Quali posizioni degli scambi possono portare a un incidente tra i tram?

- A) DESTRA-DESTRA-DESTRA-DESTRA
- B) DESTRA-DESTRA-SINISTRA-SINISTRA
- C) SINISTRA-DESTRA-SINISTRA-DESTRA
- D) DESTRA-SINISTRA-DESTRA-SINISTRA



Soluzione

La risposta corretta è B). Il primo tram (che svolta a destra) si scontra con il terzo tram (che svolta a sinistra). Analogamente, il secondo tram (che svolta a destra) si scontra con il quarto tram (che svolta a sinistra).

Per ogni altra soluzione, i tram percorrono le rotaie senza alcuna collisione: nella risposta A) vengono percorse tutte le curve “esterne” all’incrocio, nelle risposte C) e D) i tram svoltano sempre lungo la stessa strada in modo parallelo. In queste soluzioni è importante notare che le curve interne non collidono, ciò che viene detto svolta “tangenziale” o “americana” a sinistra.

Questa è l’informatica!

I binari del tram – in particolare in materia di incroci – sono un esempio di risorse condivise da molteplici attori. In un certo momento, solo un tram può trovarsi in un determinato punto . . . altrimenti c’è una collisione. Il coordinamento e l’ottimizzazione della condivisione delle risorse da parte di processi paralleli è un ambito importante dell’informatica. Come con i tram, si deve essere consapevoli dei potenziali rischi e trovare regole per l’accesso sicuro, per esempio, a basi di dati o a periferiche condivise. In caso contrario, potrebbe esserci una perdita di dati o la distruzione di componenti hardware.

Nel caso dell’incrocio è una buona idea consentire solo impostazioni “sicure” degli scambi. Una buona soluzione potrebbe essere anche l’utilizzo di segnali luminosi. Anche nell’informatica si utilizzano dei veri e propri segnali, i cosiddetti “semafori”. Essi indicano se alcune risorse sono già in uso e impongono al processo che vuole accedervi di aspettare il proprio turno.

Siti web e parole chiave

incrocio di tram, sincronizzazione (di processi), accesso a risorse limitate

- <https://it.wikipedia.org/wiki/Sincronizzazione>
- <http://www.swisseduc.ch/informatik/infotraffic/logictraffic/> (tedesco)



11. La codifica delle bandiere

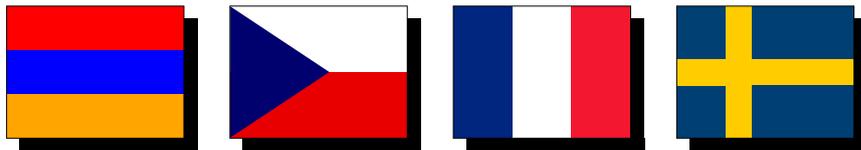
Le immagini del computer sono composte da righe di elementi di immagine (pixel). Nel caso più semplice tali immagini vengono memorizzate come file in cui il colore di ogni pixel è descritto singolarmente. Con il formato per file GIW, le immagini vengono anche compresse e quindi memorizzate in un file di minore dimensione. Il GIW (un formato di pura fantasia) funziona in questo modo:

- Ogni riga di pixel viene descritta singolarmente.
- Ogni colore è descritto attraverso un'abbreviazione di tre lettere.
- Una serie di pixel aventi lo stesso colore è descritta da una coppia di parentesi, contenente il simbolo del colore e il numero di pixel nella serie.

Per esempio, una riga di pixel descritta dalle parentesi (ver, 20) (bia, 13) è formata da 20 pixel verdi e 13 pixel bianchi.

Qui sotto potete osservare quattro immagini di bandiere. Tutte le immagini possiedono lo stesso numero di righe, ognuna delle quali possiede lo stesso numero di pixel. Tali immagini vengono memorizzate in formato GIW.

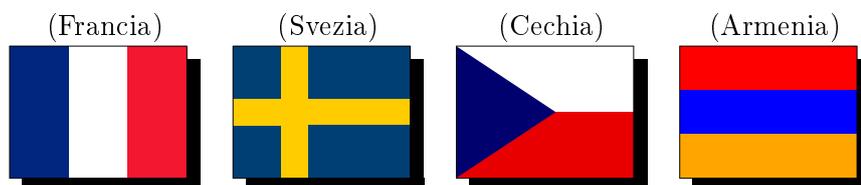
Ordina le immagini per grandezza del file GIW. Trascina le bandiere in modo che a sinistra ci sia la bandiera con il file GIW più grande.





Soluzione

La successione di bandiere esatta è:



Se una riga possiede lo stesso colore, viene descritta da una singola coppia di parentesi. Per ogni successivo cambio di colore all'interno di una riga si aggiunge poi una coppia di parentesi nella codifica. Per esempio, la bandiera dell'Armenia ha lo stesso colore in ogni riga e quindi ogni riga sarà codificata con una singola coppia di parentesi.

La bandiera della Cechia, invece, cambia colore 1 volta per ogni riga: dapprima passa dal blu al bianco e poi dal blu al rosso. Il file GIW ha dunque bisogno di 2 coppie di parentesi per ogni riga. La bandiera francese è rappresentata da 3 serie di colori per ogni riga e dunque necessita di 3 coppie di parentesi.

La bandiera svedese ha una striscia gialla orizzontale e per questa zona ha bisogno di una sola coppia di parentesi per ogni riga. Nel resto della bandiera il colore cambia 2 volte, ci vogliono quindi 3 coppie di parentesi per codificare ogni riga. Da questo si deduce facilmente che la bandiera svedese utilizza più parentesi di quella armena e meno di quella francese.

Non ci resta quindi che confrontare la bandiera svedese con quella ceca. Se la bandiera svedese avesse esattamente lo stesso numero di righe nella striscia gialla orizzontale come nella parte rimanente, avremmo bisogno in media di 2 coppie di parentesi per ogni riga. La striscia orizzontale è però chiaramente più piccola del resto della bandiera, dunque possiamo concludere che in media abbiamo bisogno più di 2 coppie di parentesi per ogni linea e affermare quindi che la bandiera svedese necessita di un file GIW più grande di quella ceca.

Questa è l'informatica!

La compressione dei dati rappresenta un ambito importante dell'informatica. Grazie alla riduzione dello spazio di memoria necessario, si possono trasportare i dati digitali attraverso la rete senza perdite di tempo. Gli algoritmi di compressione dei dati possono velocizzare notevolmente la comunicazione. Se, ad esempio, si ascoltasse la musica di una web radio senza compressione dei dati, la larghezza di banda necessaria sarebbe anche 10 volte superiore. Per questo la ricerca di nuovi algoritmi di compressione è intensa, così da poter salvare foto, musica o video in modo sempre più efficiente. La compressione utilizzata nel nostro quesito appartiene al tipo RLE (Run-length encoding). Ulteriori informazioni in merito possono essere visualizzate nel seguente filmato YouTube: https://www.youtube.com/watch?v=ydpNscvym_E.

Siti web e parole chiave

codifica, compressione, algoritmi di compressione, grafica bitmap

- https://it.wikipedia.org/wiki/Grafica_raster
- https://it.wikipedia.org/wiki/Compressione_dei_dati
- https://it.wikipedia.org/wiki/Run-length_encoding
- https://www.youtube.com/watch?v=ydpNscvym_E



12. Lacci da scarpe

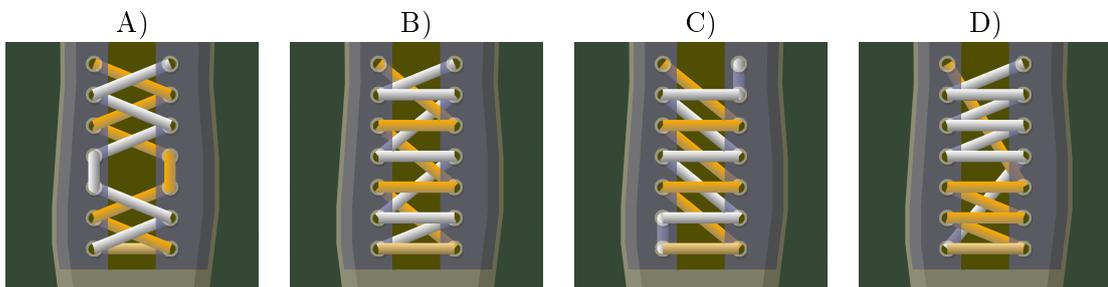
Ai castori piace creare delle allacciature della scarpa alla moda. A questo scopo, possono programmare una macchina che lega i lacci da scarpe per loro. Essa esegue i seguenti comandi:

Comando	Significato
{ ... }	Il contenuto delle parentesi viene ripetuto il più possibile.
n { ... }	Il contenuto delle parentesi viene ripetuto n volte.
arancio:	I comandi che seguono valgono solo per i lacci arancioni.
bianco:	I comandi che seguono valgono solo per i lacci bianchi.
esterno:	Il laccio passa attraverso l'occhiello, dall'interno verso l'esterno della scarpa.
interno:	Il laccio passa attraverso l'occhiello, dall'esterno verso l'interno della scarpa.
sopra:	Il laccio viene spostato verso l'alto (al prossimo occhiello).
sotto:	Il laccio viene spostato verso il basso (all'occhiello che precede).
cambio:	Il laccio viene spostato dall'occhiello di destra a quello di sinistra o viceversa.

Quale allacciatura crea il seguente programma?:

```

arancio: esterno
bianco: esterno
2{
    arancio: sopra cambio esterno
    bianco: sopra cambio esterno
}
arancio: sopra interno
bianco: sopra interno
{
    arancio: sopra cambio esterno
    bianco: sopra cambio esterno
}
    
```





Soluzione

La soluzione corretta è A). Per i primi 2 occhielli (dal basso) si applica il contenuto delle prime parentesi 2 volte, i lacci vengono poi spostati verso l'alto all'occhiello successivo. In seguito si ripetono i comandi contenuti nell'ultima parentesi fino alla fine.

Questa è l'informatica!

La macchina che crea le allacciature viene programmata con un linguaggio di programmazione semplice. Esso contiene i tipici elementi che compongono i linguaggi usati dagli informatici:

- Esecuzione sequenziale dei comandi (*Sequenze di istruzioni*)
- Ripetizione di blocchi di comandi (*cicli o loop*) e ripetizione condizionata o limitata

Ciò che manca nel linguaggio utilizzato in questo esercizio sono le *istruzioni condizionali* (selezioni), che invece vengono spesso utilizzate nella pratica, così come la creazione separata di blocchi di istruzioni funzionali (*modularizzazione*).

La macchina per le allacciature può essere anche programmata in un simulatore: <http://www.klbg.com/lacing/>. In questo caso, però, i comandi utilizzano vocaboli inglesi.

Siti web e parole chiave

cicli, comandi o istruzioni, simulatore, lacci da scarpe

- [https://it.wikipedia.org/wiki/Interprete_\(informatica\)](https://it.wikipedia.org/wiki/Interprete_(informatica))
- https://it.wikipedia.org/wiki/Linguaggio_di_programmazione
- <http://www.klbg.com/lacing/>
- <http://www.fieggen.com/shoelace/lacingmethods.htm>

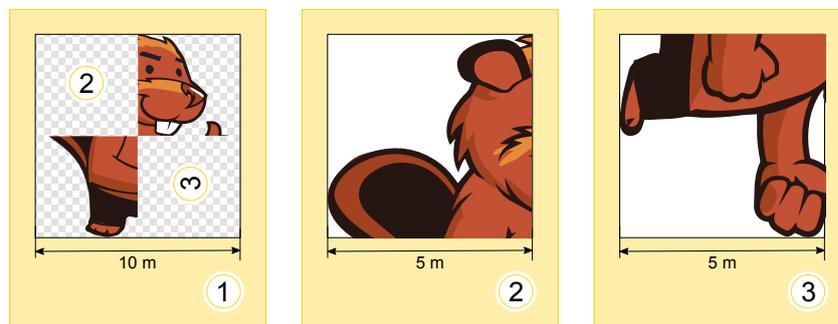


13. Pittura ricorsiva

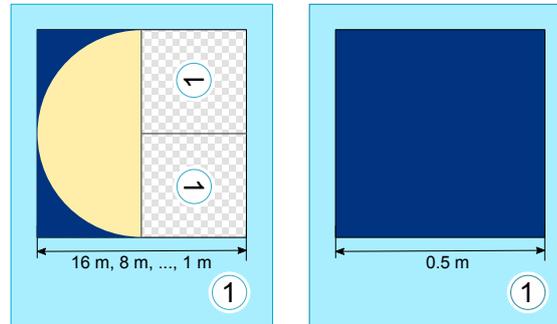
Tina e Ben aiutano a preparare una mostra nel museo di informatica. Sul pavimento di una sala d'esposizione dovrebbero dipingere un quadro di dimensioni 16 metri \times 16 metri.

Da alcuni artisti ottengono un set di schede da pittura speciali. Esse possono creare un quadro grazie a istruzioni del famoso "linguaggio dei pittori": queste istruzioni particolari contengono informazioni sugli elementi del disegno, sulle misure e sulle rotazioni. Su alcune schede da pittura sono presenti dei campi numerati che fanno riferimento ad altre schede.

Ecco un esempio tratto da un precedente progetto fatto con queste schede speciali. Se le istruzioni vengono eseguite correttamente, si ottiene il dipinto di un castoro. Si noti che le schede hanno lunghezze dei bordi diverse.

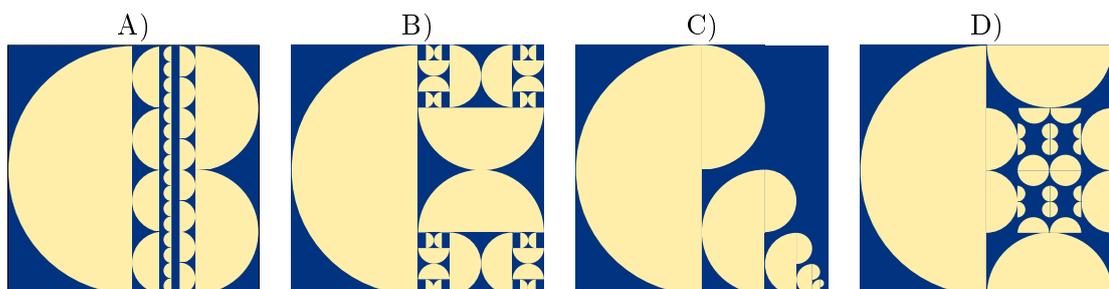


Per la sala d'esposizione Tina e Ben ricevono le seguenti schede da pittura:



Ben è confuso: "Come è possibile? La scheda a sinistra si riferisce a sé stessa, e poi le due schede hanno lo stesso numero!". Ridendo Tina dice: "Ce la faremo! Utilizziamo dapprima solo la scheda di sinistra. Useremo quella di destra in un secondo tempo, come istruzione per terminare il dipinto."

Come apparirà il pavimento della sala d'esposizione?





Soluzione

La scheda di sinistra indica che la metà sinistra del pavimento deve essere riempita con un semi-cerchio con il lato tondo orientato a sinistra. Per la metà destra, si deve utilizzare la stessa scheda due volte, ma con dimensioni ridotte. L'orientamento degli elementi dell'immagine deve essere conforme all'orientamento indicato dai campi numerati.

In tale scheda, il campo numerato posto in alto è ruotato di 90° a sinistra. Pertanto, il semi-cerchio deve essere ruotato verso sinistra e quindi il lato tondo apparirà orientato verso il basso.

In campo numerato posto in basso è invece ruotato di 90° verso destra. Pertanto, il semi-cerchio deve essere ruotato verso destra e quindi il lato tondo apparirà orientato verso l'alto.

Questa situazione si verifica solo nella risposta B). Quindi B) è l'unica risposta corretta.

Questa è l'informatica!

In informatica, gli algoritmi (ovvero insiemi di istruzioni) che richiamano sé stessi sono detti "ricorsivi". Il termine deriva dal latino "recurrere" (qui nel senso di "ritorno a sé stesso"). La ricorsione è uno strumento potente: alcuni compiti complessi possono essere risolti in modo semplice con l'utilizzo di algoritmi ricorsivi. Alcune procedure ricorsive, tuttavia, sono molto difficili da capire.

Un algoritmo ricorsivo contiene sempre una "chiamata" a sé stesso e una condizione di terminazione, che si verifica quando la ricorsione deve terminare. Se questa condizione non si verificasse, la ricorsione continuerebbe all'infinito o, almeno, fino all'esaurimento di una qualsiasi risorsa (come, ad esempio, la memoria).

Siti web e parole chiave

programmazione, ricorsione, condizione di terminazione

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



14. Giocando con le forme

Alicia si diverte con un gioco basato sulle forme geometriche. In esso, crea delle sequenze di forme in passi successivi, utilizzando ogni volta una serie di regole di sostituzione. Un esempio di regole di sostituzione è:



Alicia parte sempre da una singola forma. Ad esempio, iniziando con un quadrato e utilizzando le regole di sostituzione sopra descritte, i primi 3 passi sarebbero:



In un'ulteriore sessione di gioco, Alicia ottiene la seguente sequenza di forme utilizzando altre regole:



Quali regole di sostituzione ha utilizzato in questa sessione di gioco?

- | | | | |
|---|---|---|---|
| <p>A)</p> <p> $\square \rightarrow \triangle \square \square$
 $\triangle \rightarrow \circ$
 $\circ \rightarrow \triangle \triangle$ </p> | <p>B)</p> <p> $\square \rightarrow \square \circ \circ$
 $\triangle \rightarrow \triangle \square$
 $\circ \rightarrow \square \triangle$ </p> | <p>C)</p> <p> $\triangle \rightarrow \triangle \triangle$
 $\square \rightarrow \circ \circ$
 $\circ \rightarrow \triangle \square \triangle$ </p> | <p>D)</p> <p> $\triangle \rightarrow \square \circ$
 $\square \rightarrow \triangle \triangle \triangle$
 $\circ \rightarrow \square \triangle$ </p> |
|---|---|---|---|



Soluzione

La risposta corretta è B). Iniziando con un triangolo, Alicia esegue le sostituzioni seguenti:



Le altre possibilità vengono scartate poiché:

Risposta A). Se Alicia iniziasse con un triangolo o un cerchio non potrebbe mai ottenere un quadrato. Iniziando con un quadrato, invece, otterrebbe:



Dato che ogni serie cresce in lunghezza ad ogni passo, da questa situazione è impossibile ottenere la sequenza desiderata;

Risposta C). Se Alicia iniziasse con un triangolo, non potrebbe mai ottenere né un quadrato né un cerchio. Iniziando con un quadrato, invece, otterrebbe:



e i due triangoli iniziali non possono essere trasformati in un quadrato, come nella sequenza desiderata. Infine, iniziando con un cerchio, otterrebbe:



e anche in questo caso il problema dei due triangoli iniziali non potrebbe essere risolto;

Risposta D). Non sarebbe possibile creare 2 cerchi adiacenti. In particolare, non sarebbe possibile creare più di un cerchio, dato che ogni cerchio, nel passo successivo, sarebbe sostituito da un quadrato e un triangolo. Per questo non si può ottenere la combinazione desiderata.

Questa è l'informatica!

Le regole utilizzate nel nostro esempio sono di fatto un insieme di regole di riscrittura (“string rewriting system (SRS)”) che descrivono in concreto una grammatica context-free, in cui ogni parola (qui “forma geometrica”) viene sostituita da una o più parole.

Una grammatica context-free è formata da:

- un insieme di parole o simboli (nel nostro caso forme geometriche , e). Contrariamente al nostro esempio, esistono due tipologie di parole: quelle che possono essere ulteriormente sostituite (“non terminali”) e altre che non possono esserlo più (“terminali”).
- un simbolo di inizio (nel nostro caso si può scegliere tra le forme geometriche disponibili)
- un insieme di regole di produzione (nel nostro caso le regole di sostituzione)

Contrariamente al nostro esempio, nelle grammatiche context-free non sempre vengono utilizzate tutte le regole di produzione contemporaneamente, si può invece scegliere quale regola applicare.

Le grammatiche context-free vengono utilizzate in vari contesti: da linguaggi formali, come quelli di programmazione, allo studio di lingue naturali (italiano, tedesco, ...) fino a giungere alla descrizione del processo di crescita nelle piante.

Per risolvere questo quesito, oltre a capire le regole di sostituzione, si deve anche possedere la capacità di analizzare le sequenze di simboli. Quest'ultima operazione in informatica è detta “parsing” ed è un passo fondamentale per trasformare le istruzioni scritte in un programma in comandi che il computer può comprendere ed eseguire.



Siti web e parole chiave

grammatica context-free (CFG), parsing

- https://it.wikipedia.org/wiki/Grammatica_libera_dal_contesto
- <https://en.wikipedia.org/wiki/L-system> (in inglese)
- <https://it.wikipedia.org/wiki/Parsing>





15. Scoiattoli egoisti

Gli scoiattoli vivono nelle buche degli alberi. Un albero ha cinque buche poste una sopra all'altra. Su di esso vivono sedici scoiattoli, suddivisi nelle cinque buche.

Tutte le mattine, ogni scoiattolo controlla in quale buca ci siano stati meno scoiattoli. Più precisamente, oltre che nella propria, esso controlla quanti vicini avrebbe nella buca immediatamente sopra e quanti nella buca immediatamente sotto a quella in cui vive. Per la notte successiva sceglie quindi la buca in cui avrebbe meno scoiattoli vicini. Se nelle buche prese in esame dovesse esserci lo stesso numero di vicini, esso preferirà la propria buca alle altre e la buca superiore a quella inferiore.

Per esempio, se nella notte appena trascorsa nelle buche si trovassero dall'alto in basso 5, 0, 0, 4 e 7 scoiattoli, il giorno successivo la situazione sarebbe la seguente: tutti gli scoiattoli della buca superiore si trasferiscono in quella immediatamente sotto (0 vicini sono meglio di 4); i 7 scoiattoli della buca inferiore si trasferiscono nella buca immediatamente sopra (4 vicini sono meglio di 6); infine, i 4 scoiattoli restanti si trasferiscono nella buca immediatamente sopra (0 vicini sono meglio di 3).

Se, oggi, nelle buche dall'alto verso il basso dovessero esserci 6, 3, 3, 0 e 4 scoiattoli, in quanti giorni si ritroverebbero tutti nella stessa buca?

- A) In due giorni.
- B) In tre giorni.
- C) In quattro giorni.
- D) Non saranno mai tutti nella stessa buca.





Soluzione

La soluzione corretta è: in tre giorni.

Oggi:	6, 3, 3, 0, 4
Dopo 1 giorno:	0, 9, 0, 7, 0
Dopo 2 giorni:	9, 0, 7, 0, 0
Dopo 3 giorni:	0, 16, 0, 0, 0

Questa è l'informatica!

Questo problema è un classico esempio della swarm-intelligence (“intelligenza dello sciame”). L'idea alla base di questo algoritmo è che si possano risolvere problemi complessi attraverso l'unione di elementi molto semplici, sempre che se ne disponga a sufficienza. Per esempio, le formiche si comportano secondo semplici regole, indipendentemente una dall'altra. Quando ce ne sono molte, però, esse sono in grado di svolgere compiti impegnativi, come ad esempio costruire un formicaio e le sue gallerie, trovare la via ottimale verso una risorsa o tritare delle foglie intere.

Anche nel nostro caso abbiamo una certa quantità di elementi (gli scoiattoli) che si comportano secondo semplici regole. Il loro comportamento collettivo è però ben lungi dall'essere “intelligente”. Essi desiderano avere più spazio possibile, ma alla fine si ritroveranno tutti nella stessa buca. Da ciò ne consegue che il comportamento di uno sciame non può essere trasformato – uno a uno – in un algoritmo: a volte è infatti meglio cooperare che comportarsi egoisticamente.

Siti web e parole chiave

swarm intelligente (“intelligenza dello sciame”), algoritmi ant-colony (“colonia di formiche”)

- https://it.wikipedia.org/wiki/Swarm_intelligence
- https://it.wikipedia.org/wiki/Intelligenza_collettiva
- https://it.wikipedia.org/wiki/Algoritmo_delle_colonie_di_formiche



A. Autori dei quesiti

 Nursultan Akhmetov
 Wilfried Baumann
 Andrea Brabcová
 Eugenio Bravo
 Nicolas Brunner
 Valentina Dagienė
 Christian Datzko
 Susanne Datzko
 Olivier Ens
 Hanspeter Erni
 Gerald Futschek
 Haris Gavranovic
 Martin Guggisberg

 Hans-Werner Hein
 Filiz Kalelioğlu
 Akiko Kikui
 Hiroki Manabe
 Khairul A. Mohamad Zaki
 Hamed Mohebbi
 Anna Morpurgo
 Tom Naughton
 Erkulan Nurtazanov
 Henry Ong
 Serena Pedrocchi
 Wolfgang Pohl
 Ilya Posov

 Sergei Pozdniakov
 J. P. Pretti
 Lorenzo Repetto
 Kirsten Schlüter
 Eljakim Schrijvers
 Seiichi Tani
 Willem van der Vegt
 Jiří Vaníček
 Troy Vasiga
 Lina Vinikienė
 Michael Weigend



B. Sponsoring: concorso 2016

HASLERSTIFTUNG

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

ROBOROBO

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

digitec.ch

<http://www.digitec.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



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/>



ITgirls@hslu

<https://www.hslu.ch/de-ch/informatik/agenda/veranstaltungen/fuer-schulen/itgirls/>
HLSU, Lucerne University of Applied Sciences and Arts
Engineering & Architecture

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 Informatikunterricht der ETH Zürich.



C. Ulteriori offerte

010100110101011001001001
0100000100101110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001

SSII

www.svia-ssie-ssii.ch
schweizerischervereinfürinformatikind
erausbildung//sociétéssuisse 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.