



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

Quesiti e soluzioni 2017 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
01001001010010010010001

SSI

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



Hanno collaborato al Castoro Informatico 2017

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

Un particolare ringraziamento va a:

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

Valentina Dagiene: Bebras.org

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

Anna Morpurgo, Violetta Lonati, Mattia Monga: Italia

Gerald Futschek, Wilfried Baumann: Austrian Computer Society, Austria

Zsuzsa Pluhár: ELTE Informatikai Kar, Ungheria

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

Roman Hartmann: hartmannGestaltung (Flyer Castoro Informatico Svizzera)

Christoph Frei: Chragokyberneticks (Logo Castoro Informatico Svizzera)

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

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

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

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



INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

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

HASLERSTIFTUNG

Nota: Tutti i link sono stati verificati l'01.11.2017. Questo quaderno è stato creato il 9 ottobre 2019 col sistema per la preparazione di testi L^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 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>














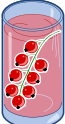



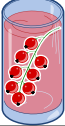



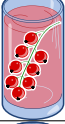
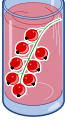



Indice

Hanno collaborato al Castoro Informatico 2017	i
Premessa	ii
1. Succhi di frutta	1
2. Sostituzioni	5
3. Esci dal labirinto!	7
4. Impianto di irrigazione	9
5. Una nuova canzone	11
6. Pista per biglie	13
7. I tunnel della diga	17
8. Aiuta l'Arabot	21
9. Gioco con stuzzicadenti	23
10. La distanza tra parole	25
11. Raccolta di punti	27
12. Metodo dei quarti	31
13. Scorciatoia o strada più lunga?	33
14. Display digitale	35
15. Suddivisione del codice	37
A. Autori dei quesiti	39
B. Sponsoring: concorso 2017	40
C. Ulteriori offerte	42



1. Succhi di frutta

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

				
Anna				
Beat				
Christine				
Daniel				

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



Soluzione

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

Anna				
Beat				
Christine				
Daniel				

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

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

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

Questa è l'informatica!

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

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

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



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

Parole chiave e siti web

Ottimizzazione, accoppiamento (matching)

- [https://it.wikipedia.org/wiki/Ottimizzazione_\(matematica\)](https://it.wikipedia.org/wiki/Ottimizzazione_(matematica))
- https://it.wikipedia.org/wiki/Branch_and_bound
- [https://it.wikipedia.org/wiki/Accoppiamento_\(teoria_dei_grafi\)](https://it.wikipedia.org/wiki/Accoppiamento_(teoria_dei_grafi))





2. Sostituzioni

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

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

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



Soluzione

La risposta corretta è D).

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

Questa è l’informatica!

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

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

Parole chiave e siti web






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

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



3. Esci dal labirinto!

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

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

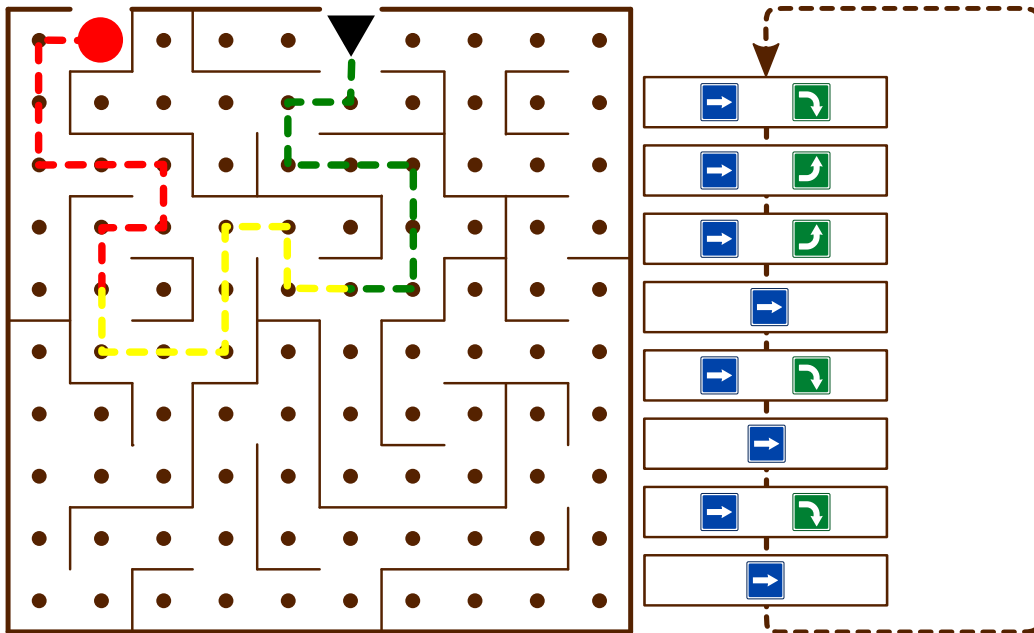
Anche se Luca può ricordarsi solo un massimo di otto istruzioni, può comunque ripeterle in sequenza più volte fino all'uscita.

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



Soluzione

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



Questa è l'informatica!

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

Parole chiave e siti web

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

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

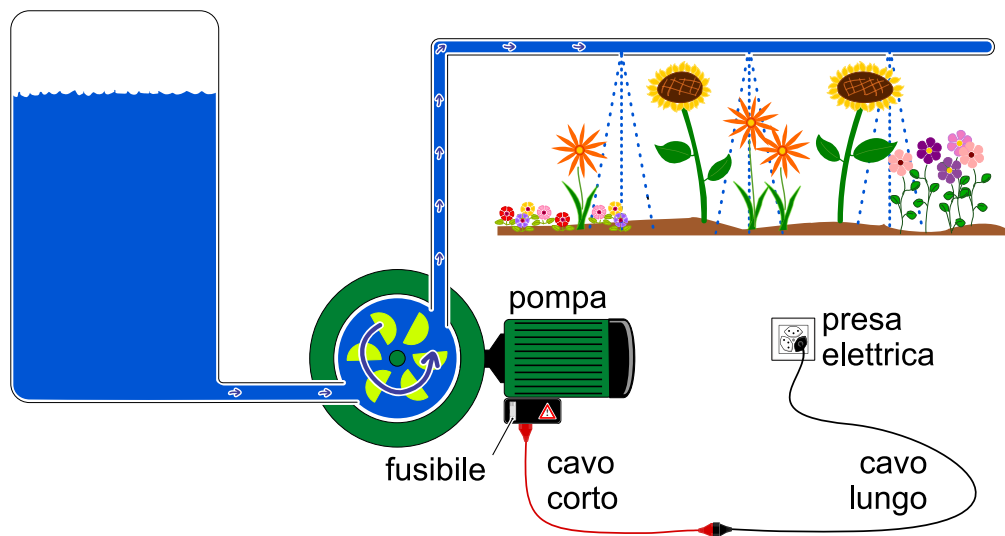


4. Impianto di irrigazione

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

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

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



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

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

Indica quali delle seguenti affermazioni sono corrette.

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



Soluzione

Le risposte corrette sono B) e C).

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

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

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

Questa è l'informatica!

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

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

Parole chiave e siti web

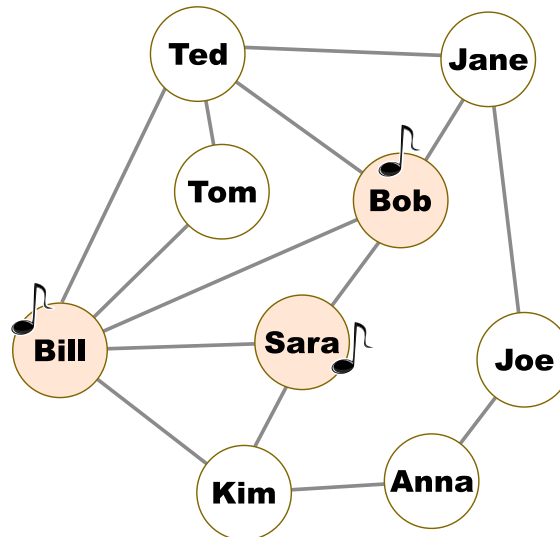
Debugging, verifica del software

- <https://en.wikipedia.org/wiki/Systemantics>
- https://en.wikiquote.org/wiki/John_Gall
- <https://it.wikipedia.org/wiki/Debugging>



5. Una nuova canzone

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



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

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

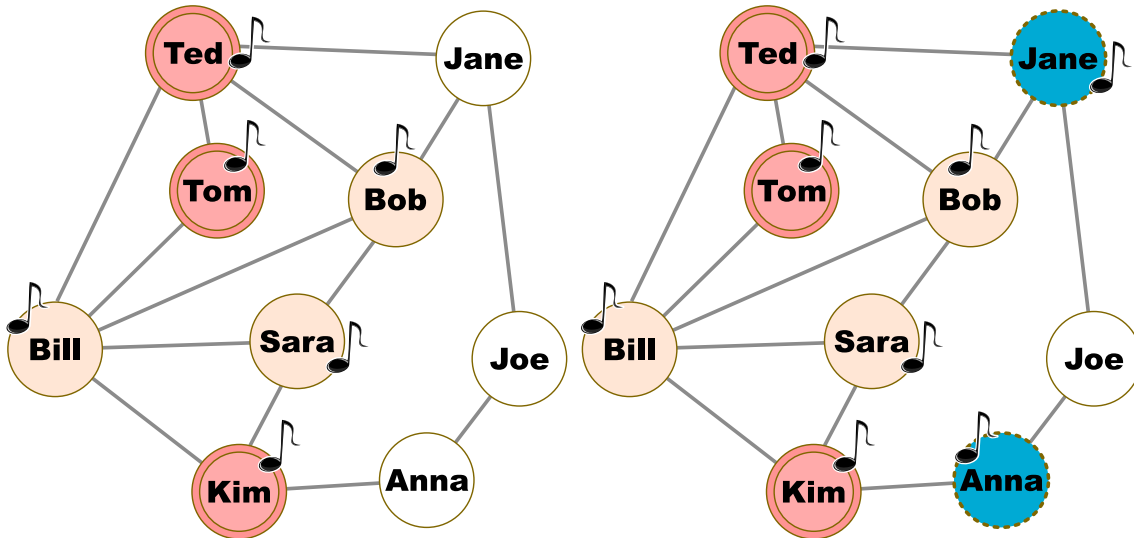
- A) Mercoledì
- B) Giovedì
- C) Venerdì
- D) Sabato



Soluzione

La risposta corretta è B).

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



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

Questa è l'informatica!

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

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

Parole chiave e siti web

Social network, grafo

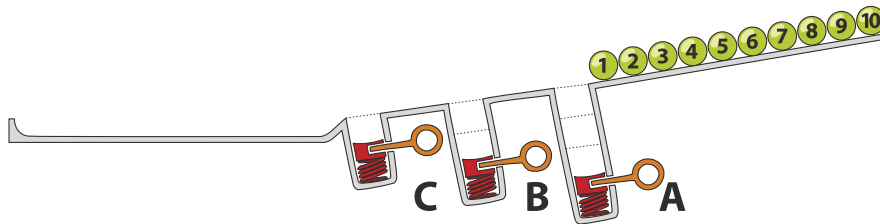
- https://it.wikipedia.org/wiki/Teoria_del_mondo_piccolo



6. Pista per biglie

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

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



In quale sequenza troveremo le biglie alla fine?

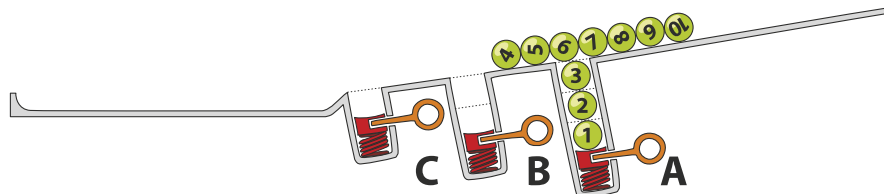
- A)  B)  C)  D) 



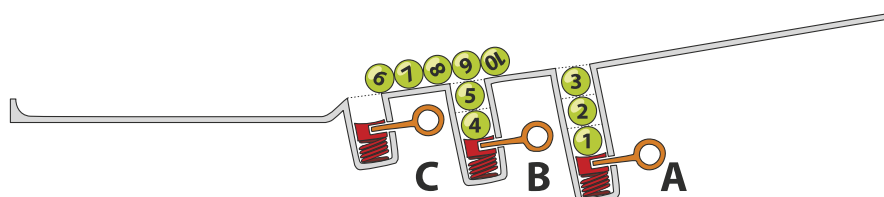
Soluzione

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

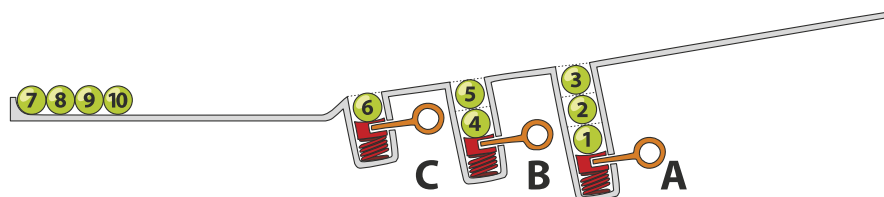
Le biglie 1, 2 e 3 cadono nella buca A.



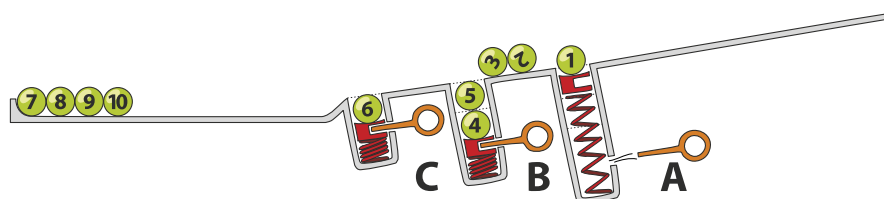
Le biglie 4 e 5 superano la buca A (ormai piena) e cadono nella buca B.



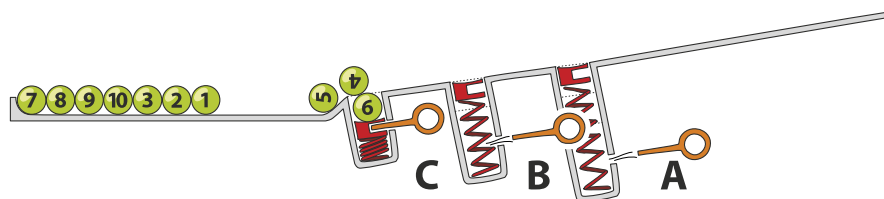
La biglia 6 cade nella buca C. Le biglie da 7 a 10 giungono invece alla fine della rampa.



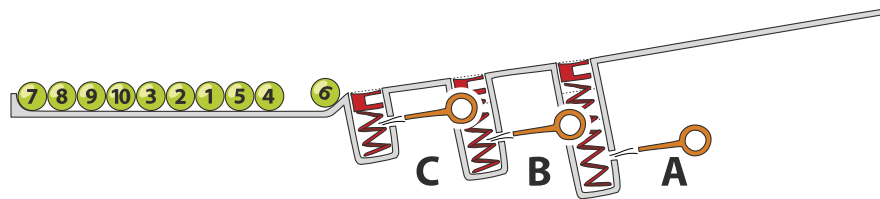
Dopo il rilascio della molla nella buca A, le biglie 3, 2 e 1 arrivano alla fine della rampa in ordine invertito rispetto alla partenza. La sequenza temporanea è quindi 7, 8, 9, 10, 3, 2, 1.



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



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



Questa è l'informatica!

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

Parole chiave e siti web

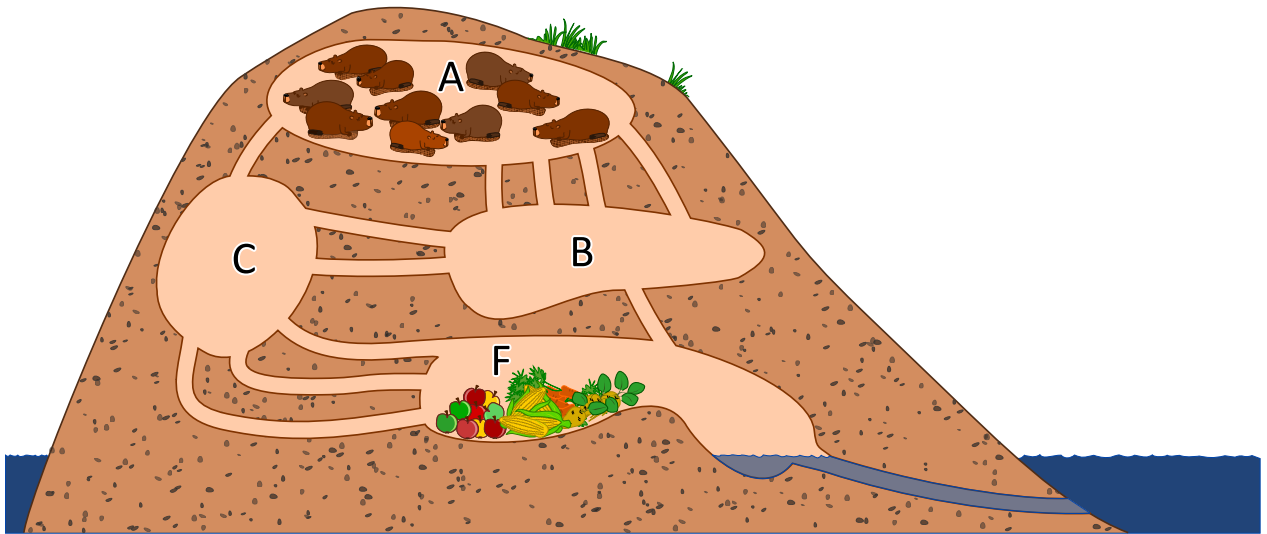
stack (pila), LIFO

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





7. I tunnel della diga



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



Soluzione

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

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

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

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

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

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

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

Questa è l'informatica!

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



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

Parole chiave e siti web

grafo, flusso di rete

- https://it.wikipedia.org/wiki/Rete_di_flusso
- [https://it.wikipedia.org/wiki/Digrafo_\(matematica\)](https://it.wikipedia.org/wiki/Digrafo_(matematica))
- https://it.wikipedia.org/wiki/Algoritmo_di_Ford-Fulkerson

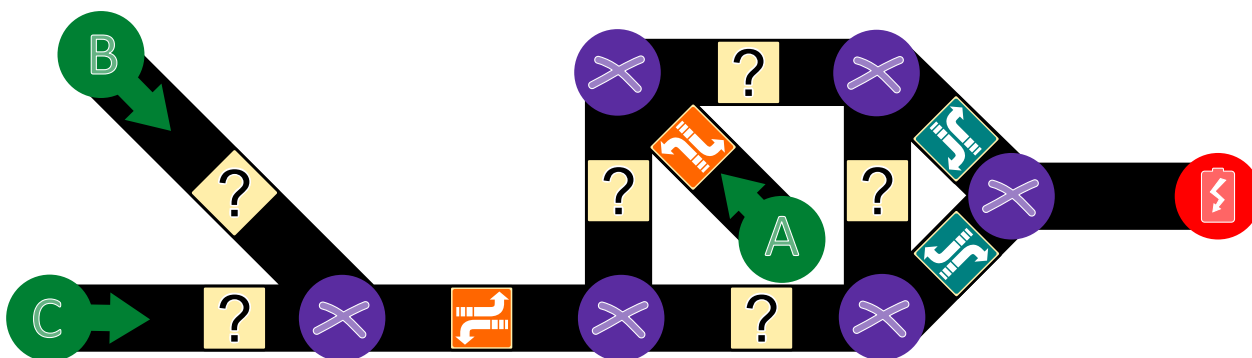




8. Aiuta l'Arabot

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

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

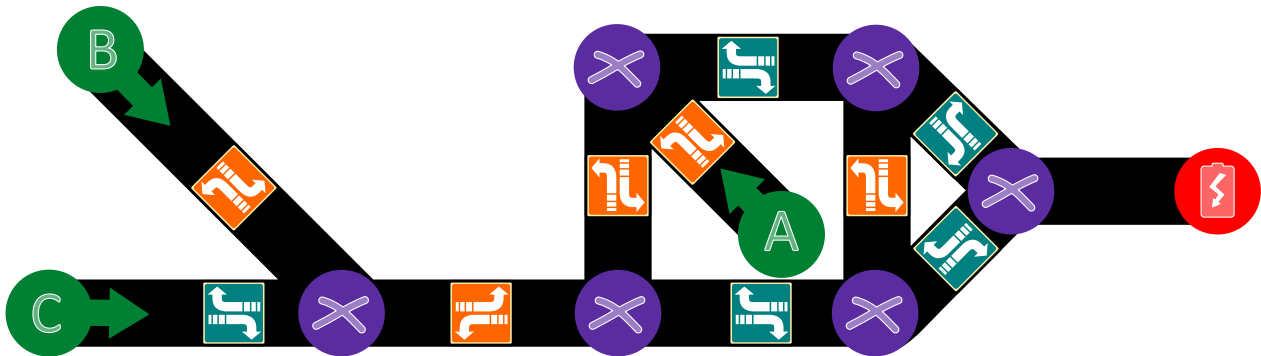




Aiuta Jonas a trovare le giuste indicazioni per tutte le linee.








Soluzione





La risposta corretta è:



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

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

Questa è l'informatica!

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

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

Parole chiave e siti web

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

- https://it.wikipedia.org/wiki/Teoria_della_complessità_computazionale
- https://en.wikipedia.org/wiki/Computational_biology
- https://en.wikipedia.org/wiki/In_silico_medicine



9. Gioco con stuzzicadenti

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

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

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

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

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



Soluzione

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

Questa è l'informatica!

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

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

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

Parole chiave e siti web

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

- https://it.wikipedia.org/wiki/Gioco_astratto
- https://it.wikipedia.org/wiki/Gioco_in_forma_estesa
- https://en.wikipedia.org/wiki/Combinatorial_game_theory



10. La distanza tra parole

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

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

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

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

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

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



Soluzione

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

Emil → Eril → Eric → Erich

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

Questa è l’informatica!

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

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

Parole chiave e siti web

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

- https://it.wikipedia.org/wiki/Distanza_di_Levenshtein
- https://en.wikibooks.org/wiki/Algorithm_Implementation/Strings/Levenshtein_distance



11. Raccolta di punti

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

	2	0	1	1	A
	1	2	0	2	3
	2	2	0	2	1
	3	1	0	2	0
↑	P	0	1	3	0
				→	

Qual è il punteggio massimo ottenibile?

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



Soluzione

La risposta corretta è C) 14.

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

2	0	1	1	A
1	2	0	2	3
2	2	0	2	1
3	1	0	2	0
0	0	1	3	0

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

2	0	1	1	A
1	2	0	2	3
2	2	0	2	1
3	1	0	2	0
0	0	1	3	0

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

2	0	1	1	A
1	2	0	2	3
2	2	0	2	1
3	4	0	2	0
0	0	1	3	0

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

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

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

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

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

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



Questa è l'informatica!

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

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

Parole chiave e siti web

Programmazione dinamica, Memoizzazione

- https://it.wikipedia.org/wiki/Programmazione_dinamica
- <https://it.wikipedia.org/wiki/Memoizzazione>





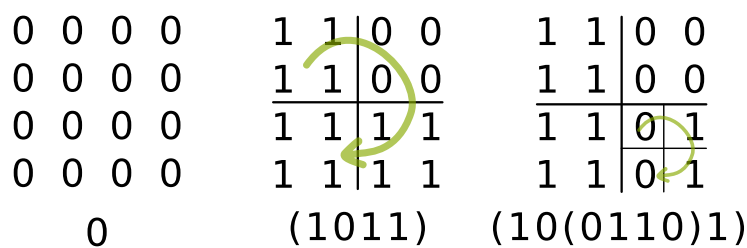
12. Metodo dei quarti

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

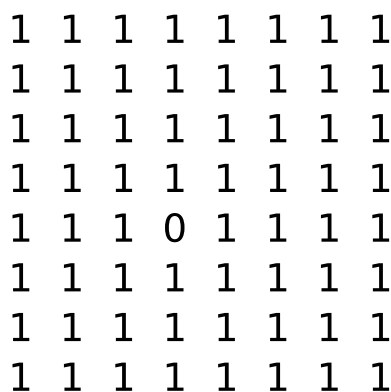
Molte immagini possono però essere rappresentate da una minore quantità di simboli, quando viene applicato il metodo dei "quarti". Con esso, i simboli binari dell'immagine vengono collocati in su una griglia quadrata. Il metodo dei quarti viene poi applicato come segue:

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

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



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



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



Soluzione

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

(111(1(1011)11))

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

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

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

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

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

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

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

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

Questa è l'informatica!

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

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

Parole chiave e siti web

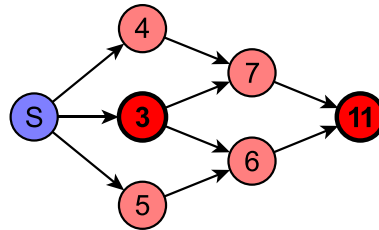
Quadtree, compressione di immagini

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



13. Scorciatoia o strada più lunga?

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



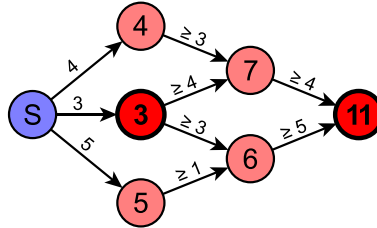
Quale delle seguenti affermazioni è vera per gli incroci segnati in rosso?

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



Soluzione

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



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

Questa è l’informatica!

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

Parole chiave e siti web

cammino minimo

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



14. Display digitale

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



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

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





Soluzione

La risposta corretta è:



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



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

Questa è l'informatica!

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

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

Parole chiave e siti web

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

- https://it.wikipedia.org/wiki/Display_a_sette_segmenti
- [https://it.wikipedia.org/wiki/Ridondanza_\(ingegneria\)](https://it.wikipedia.org/wiki/Ridondanza_(ingegneria))



15. Suddivisione del codice

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

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

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

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



Soluzione

La risposta corretta è:

1 21 1 22 33 321

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

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

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

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

Questa è l’informatica!

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

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

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

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

Parole chiave e siti web

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

- https://it.wikipedia.org/wiki/Codice_prefisso
- https://it.wikipedia.org/wiki/Codifica_di_Huffman



A. Autori dei quesiti

 Andrea Adamoli	 Yasemin Gülbahar	 Sergei Pozdniakov
 Wilfried Baumann	 Martin Guggisberg	 J.P. Pretti
 Bartosz Bieganski	 Urs Hauser	 Frances Rosamond
 Daphne Blokhuis	 Juraj Hromkovič	 Kirsten Schlüter
 Eugenio Bravo	 Ungyeol Jung	 Victor Schmidt
 Carmen Bruni	 Filiz Kalelioğlu	 Eljakim Schrijvers
 Marios Choudary	 Dong Yoon Kim	 Masood Seddighin
 Zsófia Csepregi-Horváth	 Vaidotas Kinčius	 Taras Shpot
 Valentina Dagienė	 Ivana Kosírová	 Seiichi Tani
 Christian Datzko	 Regula Lacher	 Jiří Vaníček
 Susanne Datzko	 Milan Lukić	 Troy Vasiga
 Janez Demšar	 Dario Malchiodi	 Nicolette Venn
 Olivier Ens	 Dimitris Mavrovouniotis	 Michael Weigend
 Hanspeter Erni	 Henry Ong	 Hongjin Yeh
 Michael Fellows	 Wolfgang Pohl	
 Gerald Futschek	 Ilya Posov	



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



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
er Ausbildung//société suisse pour l'infor
matique dans l'enseignement//società sviz
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.