INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Quesiti e soluzioni 2017
9º e 10º anno scolastico

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

A cura di:
Andrea Adamoli, Christian Datzko, Hanspeter Erni

www.svia-ssie-ssii.ch
schweizerischervereinfürinformatikinderausbildung//sociétésuissedel'informatiquedansl'enseignement//societàsvizzeraperl'informaticanell'insegnamento

Hanno collaborato al Castoro Informatico 2017

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

Un particolare ringraziamento va a:
Juraj Hromkovič, Giovanni Serafini, Urs Hauser, Regula Lacher, Ivana Kosírová: ETHZ
Valentina Dagiene: Bebras.org
Hans-Werner Hein, Wolfgang Pohl: Bundesweite Informatikwettbewerbe (BWINF), Germania
Anna Morpurgo, Violetta Lonati, Mattia Monga: Italia
Gerald Futschek, Wilfried Baumann: Austrian Computer Society, Austria
Zsuzsa Pluhár: ELTE Informatikai Kar, Ungheria
Eljakim Schrijvers, Daphne Blokhuis: Eljakim Information Technology bv, Paesi Bassi
Roman Hartmann: hartmannGestaltung (Flyer Castoro Informatico Svizzera)
Christoph Frei: Chragokyberneticks (Logo Castoro Informatico Svizzera)
Pamela Aeschlimann, Andreas Hieber, Aram Loosmann, Daniel Vuille, Peter Zurflüh: Lernetz.ch (pagina web)
Andrea Leu, Maggie Winter, Brigitte Maurer: Senarclens Leu + Partner

L'edizione dei quesiti in lingua tedesca è stata utilizzata anche in Germania e in Austria.
La traduzione francese è stata curata da Nicole Müller mentre quella italiana da Andrea Adamoli.

INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

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

HASLERSTIFTUNG

Nota: Tutti i link sono stati verificati l'01.11.2017. Questo quaderno è stato creato il 18 novembre 2017 col sistema per la preparazione di testi LaTeX.

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.bebras.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» (3º e 4º 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:

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

Gli alunni iscritti al 3º e 4º 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.

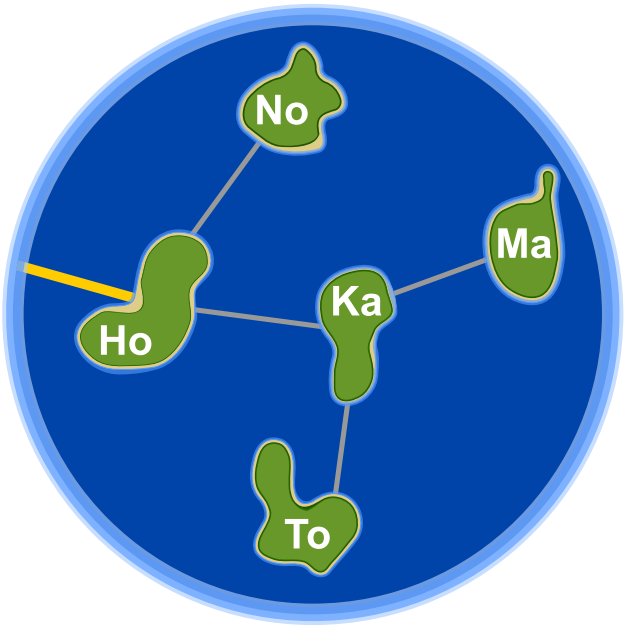
Per ulteriori informazioni:

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

1. Honomakato

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



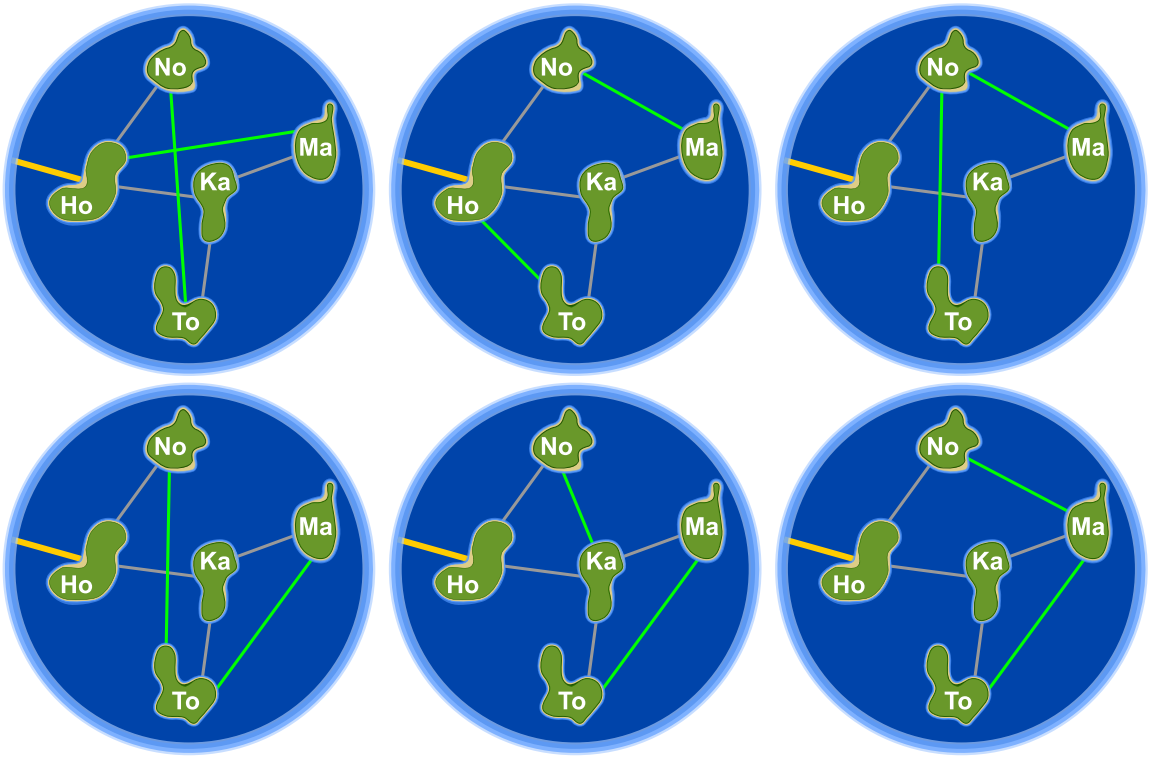
Gli abitanti di Honomakato desiderano avere una connessione affidabile a Internet per tutte le isole: in altre parole, anche se un cavo tra di esse dovesse danneggiarsi, ogni isola minore dovrebbe sempre poter accedere a Internet.
Fai in modo che ogni isola abbia un collegamento affidabile a Internet. Piazza 2 ulteriori cavi tra le isole.

Soluzione

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

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

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



Questa è l'informatica!

Le rete di cavi con la quale l'arcipelago di Honomakato è collegato a Internet rappresenta una piccola parte della rete globale e un esempio di come essa è costruita. I router, i server e altri dispositivi telematici sono i nodi di una grossa rete chiamata Internet, esattamente come le isole lo sono nel nostro arcipelago.
Internet è stato concepito negli anni '60 come rete robusta (ossia, "affidabile"). Un guasto alle singole connessioni tra i nodi non deve in alcun modo pregiudicare il funzionamento dell'intera rete.

Pertanto ogni nodo dispone di connessioni multiple e in caso di rottura o congestione di una di esse, le altre possono essere utilizzate come riserva. Anche per altre reti (come quelle di trasporto o quelle di approvvigionamento) è importante che ogni nodo non venga isolato in caso di rottura di una connessione.
L'informatica utilizza la teoria dei grafi per eseguire calcoli su questo tipo di reti. Un grafo è definito da una rete di nodi e di collegamenti (detti archi) tra essi. Un grafo è detto "connesso" se per ogni coppia di nodi A e B, B è collegato ad A da un percorso attraverso uno o più archi. Un singolo arco necessario affinché un grafo possa essere detto connesso, viene chiamato "ponte". In informatica sono stati sviluppati degli algoritmi per individuare questi ponti, in particolare Robert Tarjan ne ha sviluppato uno molto efficiente.

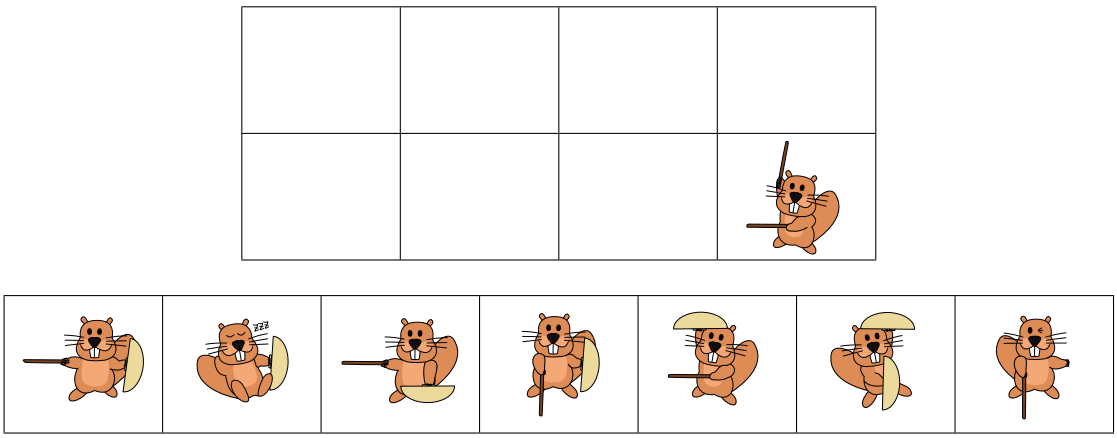
Siti web e parole chiave

Strutture di dati dinamiche, grafo, bridge ("ponte")

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

2. L'arte del bastone giapponese

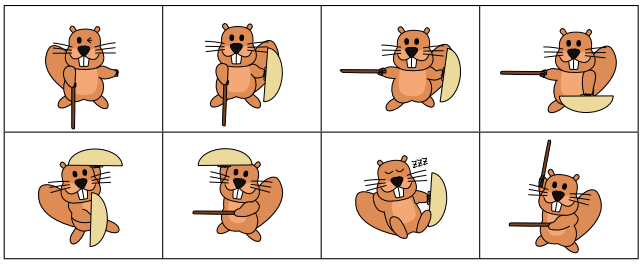
Lucia e i suoi amici fanno parte di un club che pratica l'arte del bastone giapponese. Per una foto, desiderano sedersi nel cortile scolastico in modo che ogni bastone punti verso uno scudo. Per aiutarsi a fare questo, hanno disegnato diversi campi nel cortile. Lucia si è già messa in posa e sotto a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



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

Soluzione

La risposta corretta è:



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

Questa è l'informatica!

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

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

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

Siti web e parole chiave

pensiero logico, conclusioni logiche

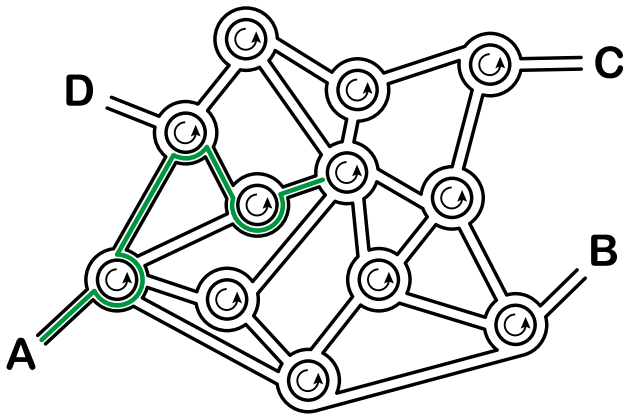
- `https://it.wikipedia.org/wiki/Backtracking`

3. La città delle rotonde

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

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

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



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

A) Conduce al punto A.

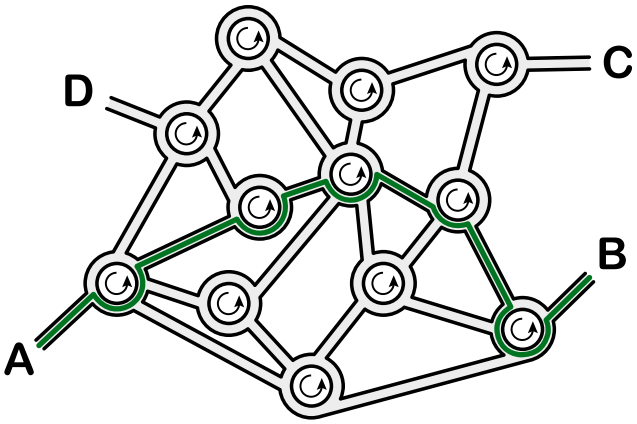
B) Conduce al punto B.

C) Conduce al punto C.

D) Conduce al punto D.

Soluzione

La spiegazione conduce al punto B.



Questa è l'informatica!

Questo compito è un buon esempio di "informazione strutturata". Un sistema di guida computerizzato non potrebbe mai comprendere le indicazioni dette nella nostra lingua. Quando, però, strutturiamo queste indicazioni ad esempio in una sequenza di numeri, le informazioni assumono una forma che il computer può facilmente interpretare. Le sequenze di istruzioni (strutturate) sono alla base di molti linguaggi di programmazione.
Nel nostro compito è utile che le rete stradale sia molto simile: tutti gli incroci, infatti, sono formati da rotonde. Strutture come queste vengono dette omogenee, in antitesi rispetto a quelle dette eterogenee. Strutture omogenee sono molto più semplici da formalizzare che le strutture eterogenee, per questo sono preferite dagli informatici.

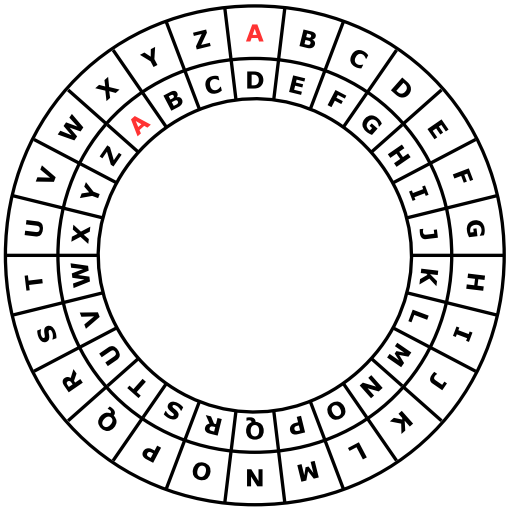
Siti web e parole chiave

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

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

4. Ordinazione segreta

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



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

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

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

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

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

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

Soluzione

La risposta corretta è LASAGNA:

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

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

Questa è l'informatica!

Anna codifica il messaggio in modo che solo il suo cuoco preferito possa capire l'ordinazione. La crittografia è un metodo utilizzato fin dai tempi antichi. Il motivo è chiaro: vogliamo essere sicuri che il messaggio sia letto solo del destinatario designato e non dalle spie. Esistono molti metodi per la crittografia, ma in generale abbiamo sempre a che fare con due tipi di algoritmi principali: quello per la codifica e quello per la decodifica. Entrambi necessitano di una chiave per eseguire il lavoro. Un metodo per la crittografia tra i più semplici risale a Giulio Cesare: in esso, la chiave è un numero che indica uno spostamento all'interno dell'alfabeto. La chiave 3, ad esempio, indica che la lettera "A" viene codificata in una "D", "B" in "E", e così via. Allo stesso tempo, indica che la lettera "D" viene de-codificata in "A", "E" in "B", ecc. Il "disco di Cesare" (meglio, "cifrario di Cesare") mostrato in questo compito aiuta nel cifrare e decifrare i messaggi con questo procedimento.
Metodi per la cifratura che usano una chiave semplice (ad esempio di un solo numero) sono relativamente insicuri. Anna lo sa bene e per questo utilizza numeri diversi per ogni lettera, utilizzando dunque un metodo simile a quello di Vigenère. In questo procedimento i numeri che formano la chiave possono essere ripetuti quanto necessario per coprire la lunghezza del messaggio e dunque la chiave non è necessariamente troppo lunga. Ma anche questo metodo può risultare insicuro, soprattuto per chiavi corte e messaggi lunghi.

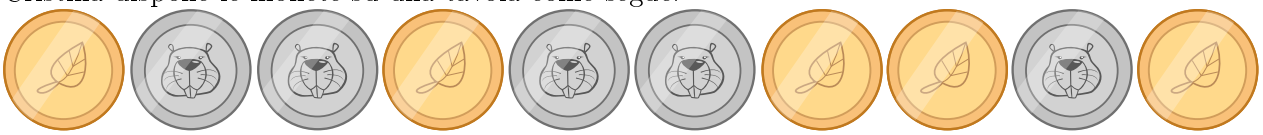
Siti web e parole chiave

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

- `https://it.wikipedia.org/wiki/Cifrario_di_Cesare`
- `https://it.wikipedia.org/wiki/Cifrario_polialfabetico`

5. Gioco di monete

Cristina possiede 10 monete con una faccia dorata () e una argentata ().
Cristina dispone le monete su una tavola come segue:



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

A) 1

B) 2

C) 4

D) 6

E) 8

F) Non è possibile.

Soluzione

Non è possibile.
Ogni volta che vengono girate due monete vicine, il numero di monete che mostrano la faccia argentata rimane sempre dispari: la "parità" delle facce delle monete non cambia.
Essendo la parità delle monete argentate sempre dispari, non mai potrà succedere che alla fine non ce ne sia almeno una.

Questa è l'informatica!

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

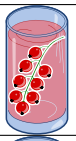
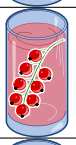
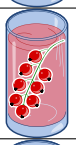
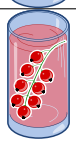
Siti web e parole chiave

parità, bit di parità

- `https://it.wikipedia.org/wiki/Bit_di_parità`

6. Succhi di frutta

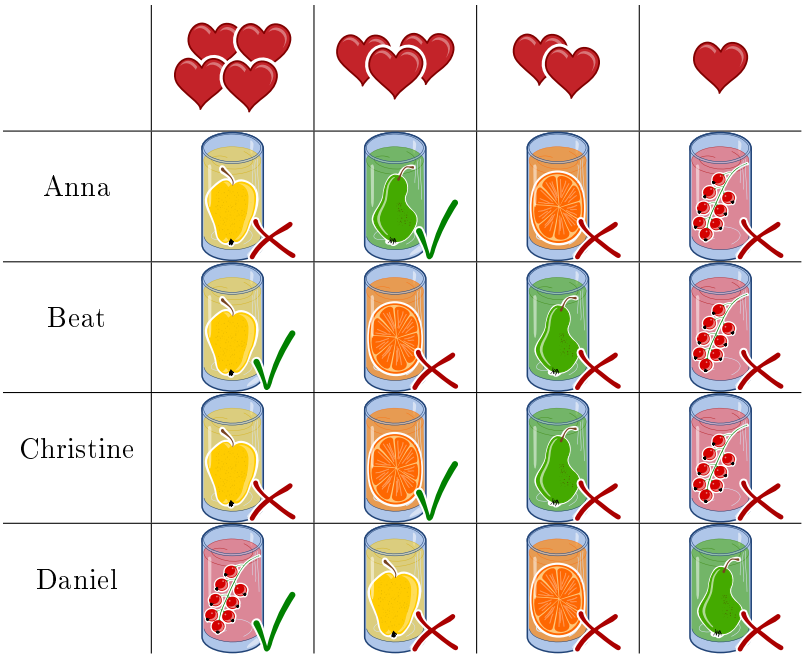
Sulla strada per le vacanze, quattro amici fanno una pausa in un negozio dove è possibile acquistare rinfrescanti succhi di frutta. Ognuno dei quattro amici ha le sue preferenze, rappresentate nella tabella qui sotto. Più cuori indicano una maggiore preferenza per la bevanda. Ad esempio, Anna valuta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



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.

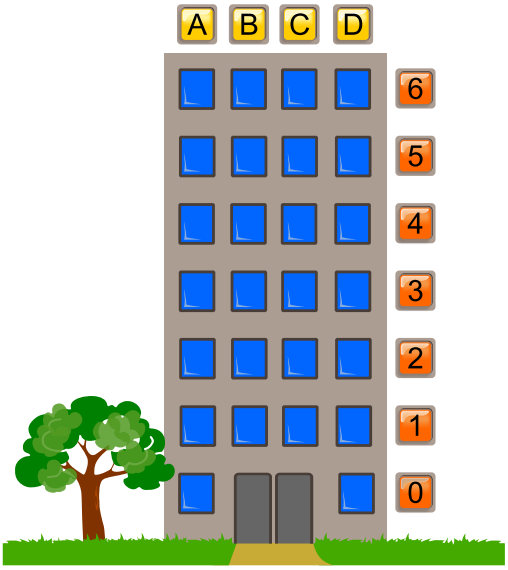
Siti web e parole chiave

Ottimizzazione, accoppiamento (matching)

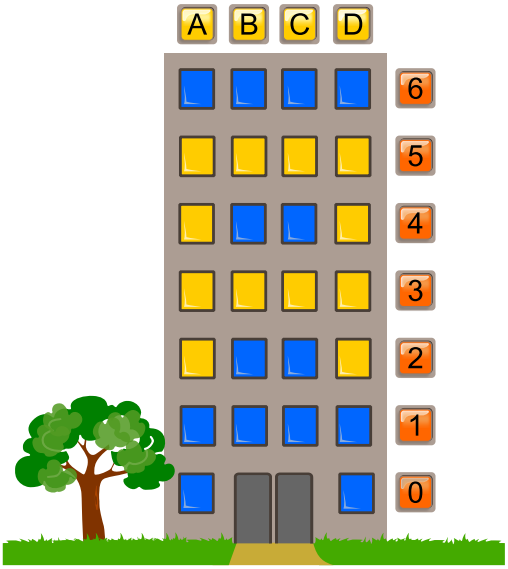
- `https://it.wikipedia.org/wiki/Ottimizzazione_(matematica)`
- `https://it.wikipedia.org/wiki/Branch_and_bound`
- `https://it.wikipedia.org/wiki/Accoppiamento_(teoria_dei_grafi)`

7. Giochi di luce

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



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



Soluzione

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

Questa è l'informatica!

Gli interruttori della luce in questo compito possono essere paragonati ai comandi di un qualsiasi sistema o macchina informatica. Anche i programmi di computer più complessi, infatti, possono essere interpretati come una sequenza di semplici comandi. I locali dietro alle finestre corrispondono a delle celle di memoria che possono avere valore 0 (luci spente) o 1 (luci accese).
I computer moderni permettono di eseguire comandi compositi che gestiscono contemporaneamente più celle di memoria. Nel nostro caso, corrispondono agli interruttori del piano o della colonna di finestre. I computer manipolano contemporaneamente milioni di celle di memoria già solo nel processore, e diventano miliardi o bilioni nella RAM e nel disco rigido.
Per questo è importante che i comandi siano definiti chiaramente, ovvero che si specifichi quando possono essere eseguiti (*pre-condition*) e cosa succede dopo (*post-condition*). Nel nostro esempio, valgono le seguenti condizioni per le luci di un piano: se anche una sola luce del piano è spenta (*pre-condition*), dopo aver premuto l'interruttore tutte le luci del piano saranno accese (*post-condition*). Altrimenti (cioè tutte le luci sono accese, *pre-condition*) tutte le luci del piano saranno spente (*post-condition*).

Siti web e parole chiave

Programmazione del hardware, sequenze di operazioni, operazioni binarie

- `https://it.wikipedia.org/wiki/Assembly`

8. 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).

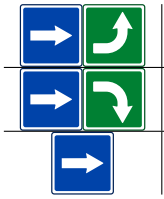
Siti web e parole chiave

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

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

9. 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:

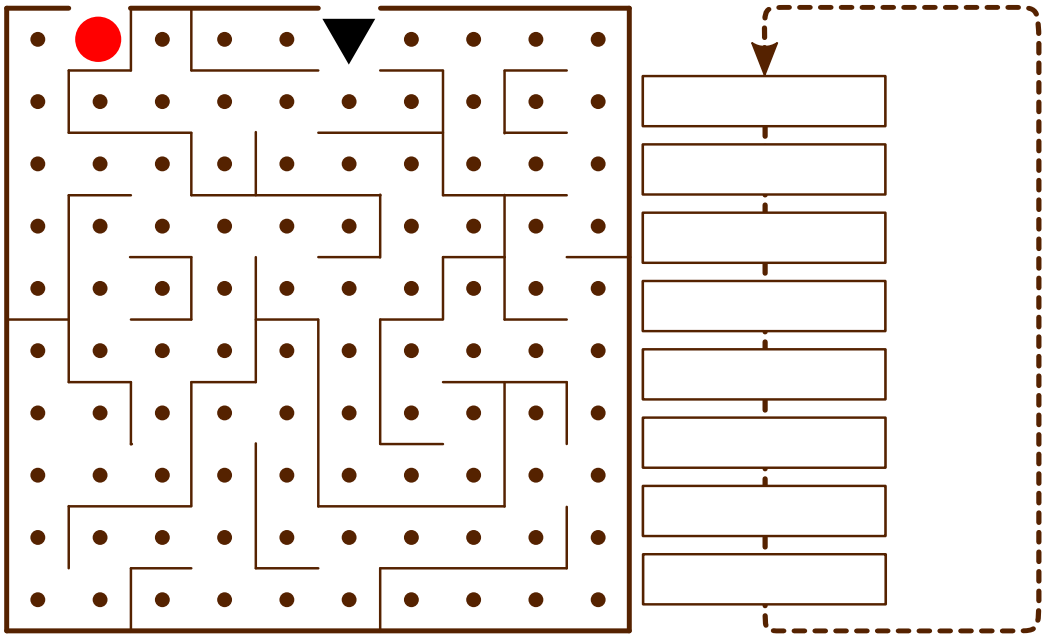


Fai un passo in avanti e girati a sinistra.

Fai un passo in avanti e girati a destra.

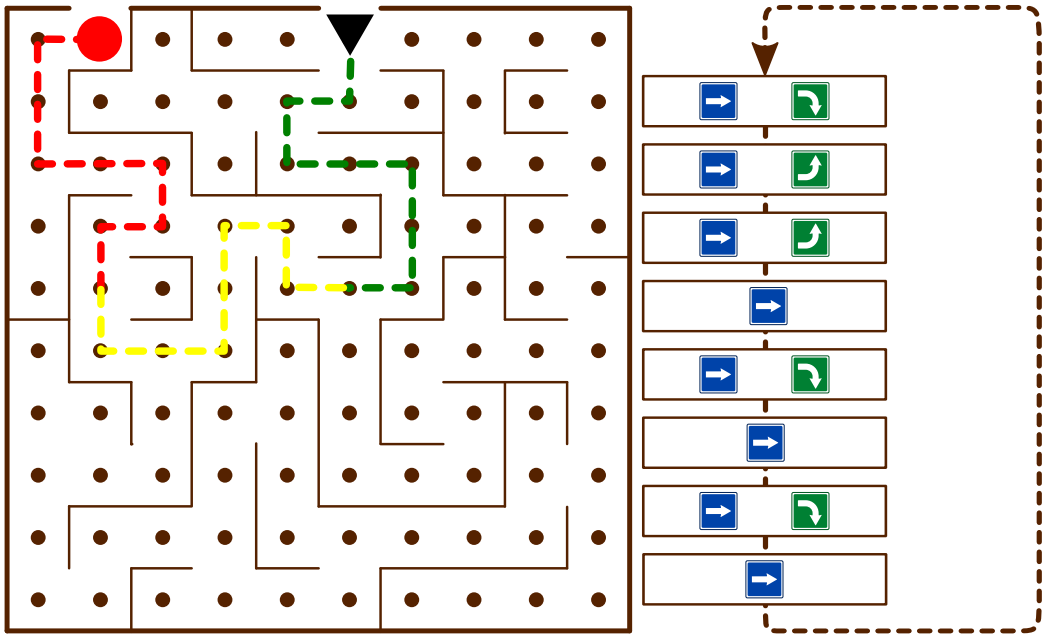
Fai un passo in avanti.

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.

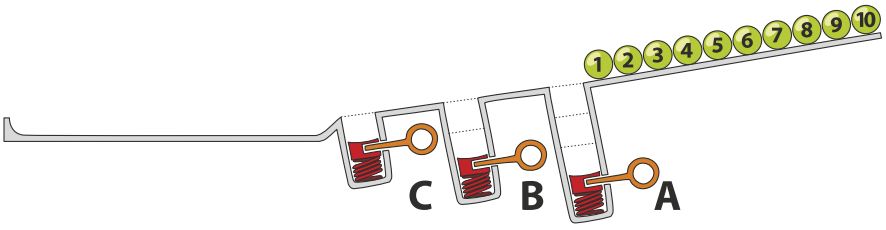
Siti web e parole chiave

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

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

10. 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) 

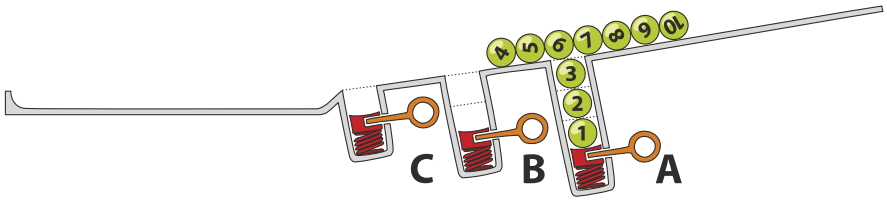
B) 

C) 

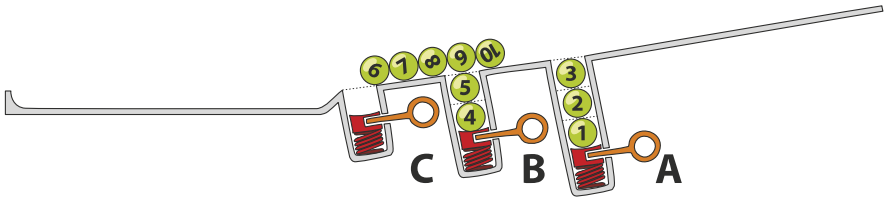
D) 

Soluzione

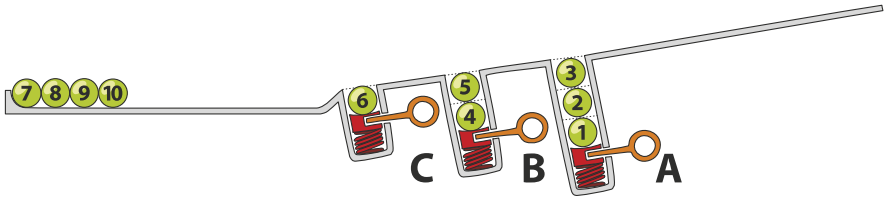
La risposta corretta è D):  .
Le biglie 1, 2 e 3 cadono nella buca A.



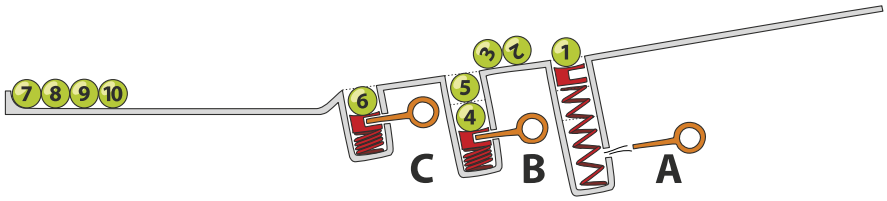
Le biglie 4 e 5 superano la buca A (ormai piena) e cadono nella buca B.



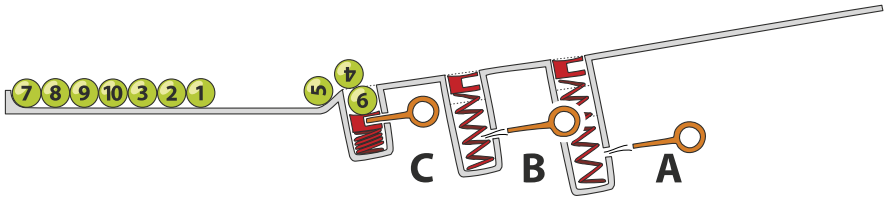
La biglia 6 cade nella buca C. Le biglie da 7 a 10 giungono invece alla fine della rampa.



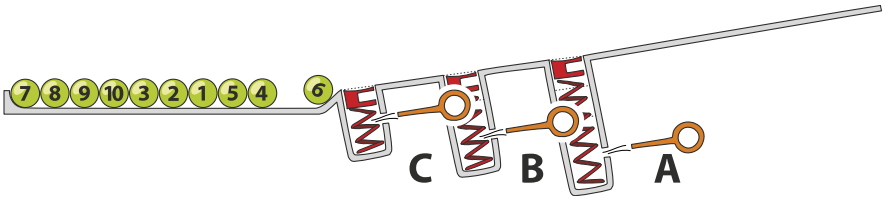
Dopo il rilascio della molla nella buca A, le biglie 3, 2 e 1 arrivano alla fine della rampa in ordine invertito rispetto alla partenza. La sequenza temporanea è quindi 7, 8, 9, 10, 3, 2, 1.



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!

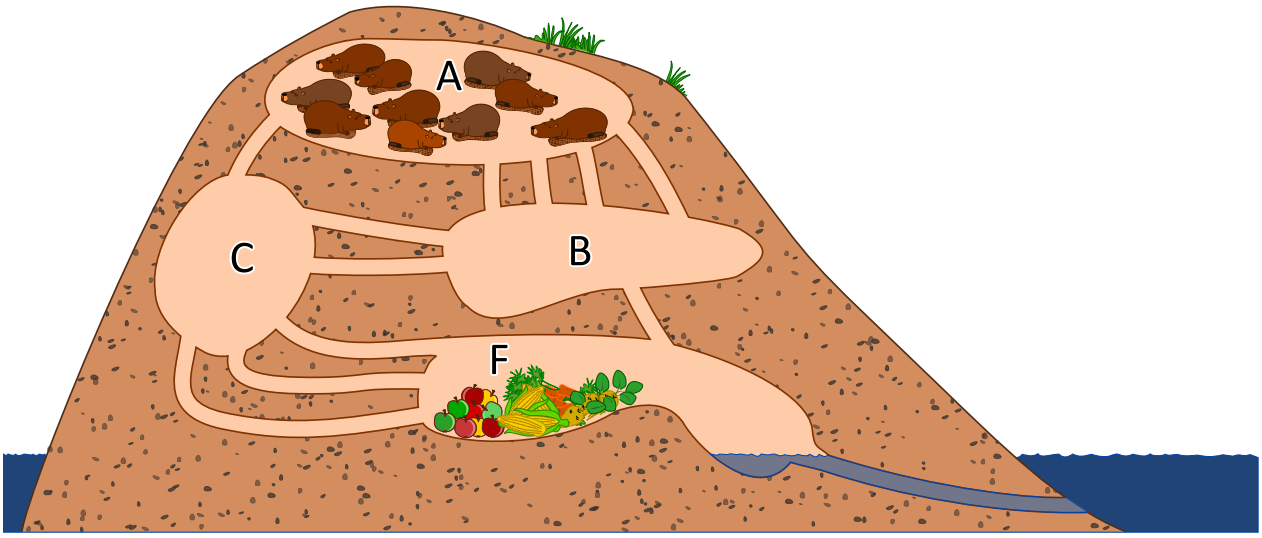
Le buche della rampa ricordano le strutture di dati chiamate pile (in inglese, *stack*). Una pila permette di salvare e poi utilizzare dei dati secondo il principio *Last-In First-Out* (*LIFO*): l'ultima biglia ad entrare in una buca è anche la prima ad uscire. Sebbene questo principio possa sembrare 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.

Siti web e parole chiave

stack (pila), LIFO

- `https://it.wikipedia.org/wiki/LIFO`
- `https://it.wikipedia.org/wiki/Pila_(informatica)`

11. 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 → B → F
- A → C → 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 → B → C → 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
3 castori vanno da A a B (non sfruttando tutti i tunnel) *1 castoro va da A a C*				
Situazione dopo 1 minuto	6	3	1	0
3 castori vanno da A a B (non sfruttando tutti i tunnel) *1 castoro va da B a F* *2 castori vanno da B a C* *1 castoro va da C a F* *1 castoro va da A a C*				
Situazione dopo 2 minuti	2	3	3	2
1 castoro va da A a B (cammino minimo) *1 castoro va da B a F* *2 castori vanno da B a C* *1 castoro va da A a C (cammino minimo)* *3 castori vanno da C a F*				
Situazione dopo 3 minuti	0	1	3	6
1 castoro va da B a F *3 castori vanno da C a F*				
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.

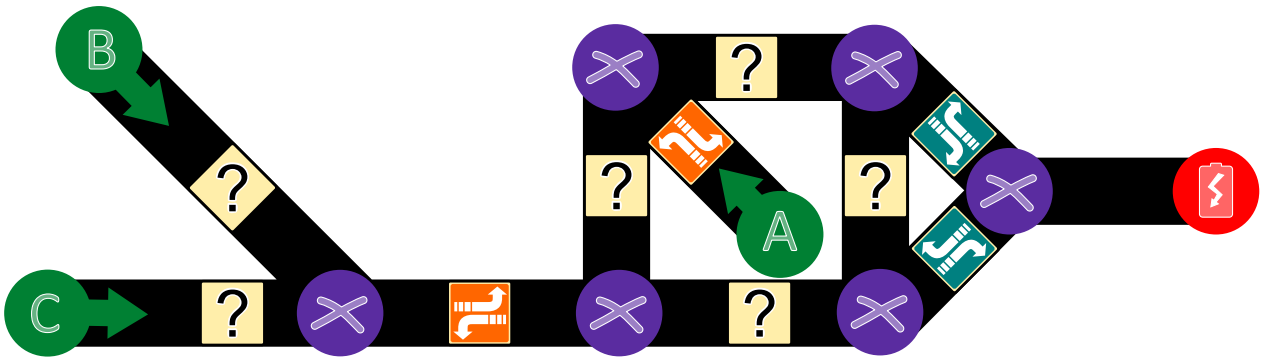
Siti web e parole chiave

grafo, flusso di rete

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

12. Aiuta l'Arabot

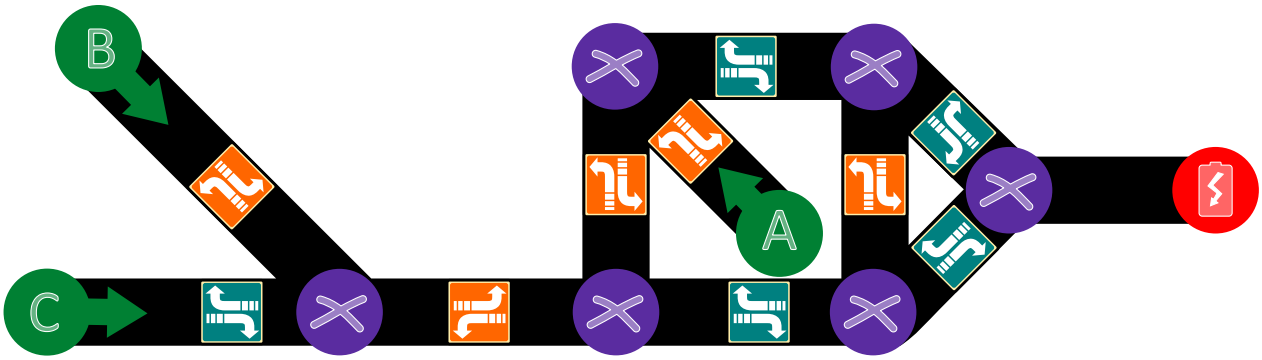
Un Arabot è un robot che si muove su un foglio di carta. Esso segue sempre le linee nere che vengono disegnate sul foglio. Su ogni linea si trova un'indicazione che gli dice se al prossimo incrocio ( ) deve prendere la prima svolta alla sua sinistra ( ) o la prima svolta alla sua destra ( ). Se dovesse 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 interessati 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*.

Siti web e parole chiave

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`

13. 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.

Siti web e parole chiave

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`

14. 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:

$$\text{Emil} \to \text{Eril} \to \text{Eric} \to \text{Erich}$$

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.

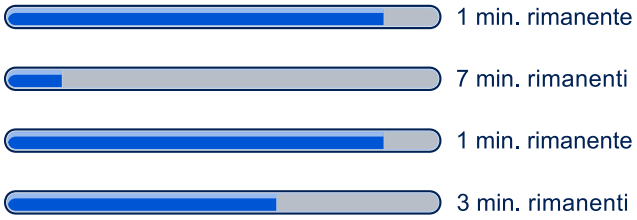
Siti web e parole chiave

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`

15. Download contemporanei

Quando si scaricano contemporaneamente più documenti di grosse dimensioni, la capacità della connessione viene condivisa. Ad esempio, se si scaricano 10 documenti, solo un decimo della velocità di connessione è disponibile per ognuno di essi.
Un utente sta scaricando 4 documenti contemporaneamente. Il tempo rimanente viene calcolato in base alla velocità attuale.



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

Soluzione

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

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

Al terzo minuto, tutti i documenti sono stati scaricati.

Questa è l'informatica!

L'*indicatore di avanzamento* (o *progress bar*) è un elemento informativo che mostra il progresso nell'elaborazione di un ordine. Esso è utilizzato ad esempio durante le installazioni i programmi o il download di documenti. Di regola, esso fornisce anche una stima del tempo rimanente. In casi particolari, quando non è possibile fornire una stima, ma si vuole segnalare che il computer sta elaborando ugualmente la richiesta si mostra un simbolo ruotante come una clessidra.
La velocità di trasmissione indica la quantità di dati digitali che viene trasmessa per ogni secondo. La velocità massima è detta capacità di un canale. Per ogni canale esistono però anche altre misure importanti, come ad esempio il ritardo nella risposta (o latenza). Essa indica il tempo impiegato da un dato per passare del trasmettitore al ricevente (a volte in questa misura è calcolato anche il tempo per il percorso inverso).
Nella rete ci sono diversi fattori che influenzano la velocità massima di trasferimento: spesso il server a cui ci si connette deve far fronte a molte richieste e la sua capacità per ogni comunicazione è quindi notevolmente ridotta da limitazioni hardware o software. I dati, poi, possono arrivarci da qualsiasi parte nel mondo e devono passare attraverso numerosi punti di instradamento (router), che gestiscono molte connessioni. La velocità è pure limitata dall'allacciamento al provider di internet attraverso il modem casalingo (di norma un decimo o anche meno della velocità massima di un server). Infine, la connessione tra modem e computer (spesso attraverso un router domestico wireless) può costituire un ennesimo collo di bottiglia. La capacità di una rete WLAN, ad esempio, è molto ridotta (più o meno un decimo) rispetto alla connessione tramite cavo "ethernet".

Siti web e parole chiave

download, indicatore di avanzamento (progress bar)

- `https://it.wikipedia.org/wiki/Progress_bar`
- `https://de.wikipedia.org/wiki/Datenübertragungsrate#Beispiele_für_Datenübertragungsraten`

A. Autori dei quesiti

Andrea Adamoli
Wilfried Baumann
Bartosz Bieganowski
Daphne Blokhuis
Eugenio Bravo
Carmen Bruni
Anton Chukhnov
Zsófia Csepregi-Horváth
Valentina Dagienė
Christian Datzko
Susanne Datzko
Janez Demšar
Olivier Ens
Hanspeter Erni
Michael Fellows
Gerald Futschek
Martin Guggisberg
Urs Hauser
Juraj Hromkovič
Filiz Kalelioğlu
Vaidotas Kinčius
Ivana Kosírová
Regula Lacher
Greg Lee
Milan Lukić
Hiroki Manabe
Mattia Monga
Henry Ong
Wolfgang Pohl
Sergei Pozdniakov
Frances Rosamond
Kirsten Schlüter
Eljakim Schrijvers
Maiko Shimabuku
Taras Shpot
Seiichi Tani
Ahto Truu
Jiří Vaníček
Troy Vasiga
Michael Weigend
Hongjin Yeh
Momo Yokoyama

B. Sponsoring: concorso 2017

http://www.haslerstiftung.ch/

http://www.roborobo.ch/

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

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

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

Standortförderung beim Amt für Wirtschaft und Arbeit Kanton Zürich

i-factory (Museo Svizzero dei Trasporti, Lucerna)

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

http://www.bbv.ch/

http://www.presentex.ch/

`http://www.phlu.ch/`
Pädagogische Hochschule Luzern

`http://www.abz.inf.ethz.ch/`
Ausbildungs- und Beratungszentrum für Informatikunterricht der ETH Zürich.

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

`https://www.zhdk.ch/`
Zürcher Hochschule der Künste

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

`http://senarclens.com/`
Senarclens Leu & Partner

C. Ulteriori offerte

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.