



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

Quesiti e soluzioni 2017 7^o e 8^o anno scolastico

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

A cura di:

Andrea Adamoli, Christian Datzko, Hanspeter Erni

010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001

SSI

www.svia-ssie-ssii.ch
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 38.



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>



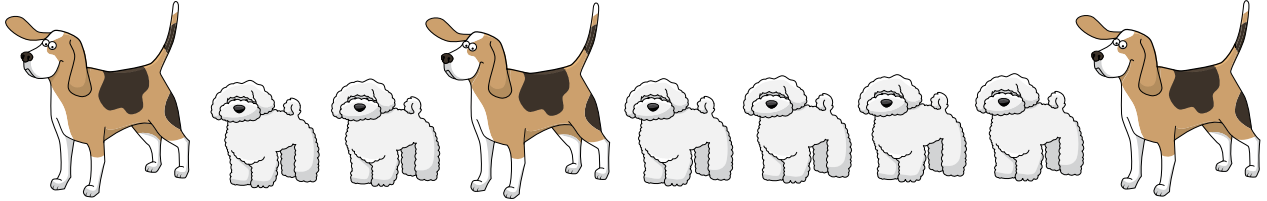
Indice

Hanno collaborato al Castoro Informatico 2017	i
Premessa	ii
1. Scambio di cani	1
2. Il nome giapponese	3
3. Un piccolo programma	5
4. Gioielli	7
5. Il giornale della scuola	9
6. Honomakato	13
7. L'arte del bastone giapponese	17
8. La pizzeria dei castori	19
9. Ordinazione segreta	21
10. Gioco di monete	23
11. Succhi di frutta	25
12. Intrusione al museo	29
13. Giochi di luce	31
14. Sostituzioni	33
15. Pista per biglie	35
A. Autori dei quesiti	38
B. Sponsoring: concorso 2017	39
C. Ulteriori offerte	41

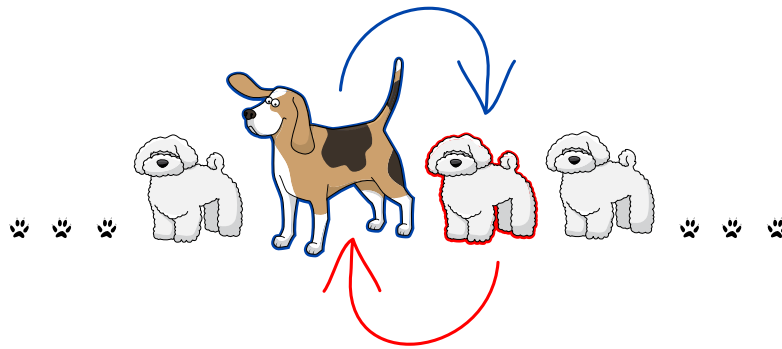


1. Scambio di cani

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



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



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

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



Soluzione

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

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

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

Questa è l'informatica!

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

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

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

Parole chiave e siti web

swap ("scambio")

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



2. Il nome giapponese

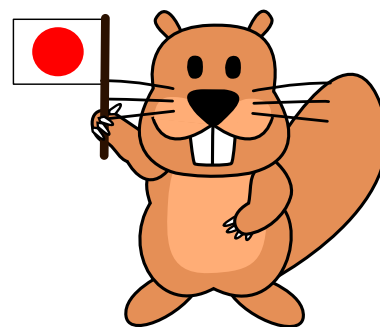
Un'amica giapponese ci racconta come, secondo una loro antica tradizione, dalle lettere del proprio nome si possa ottenere un nome spirituale, semplicemente sostituendo ogni lettera con delle sillabe associate:

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

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

Quale è il vero nome del tuo amico?

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





Soluzione

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

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

Questa è l’informatica!

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

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

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

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

Parole chiave e siti web

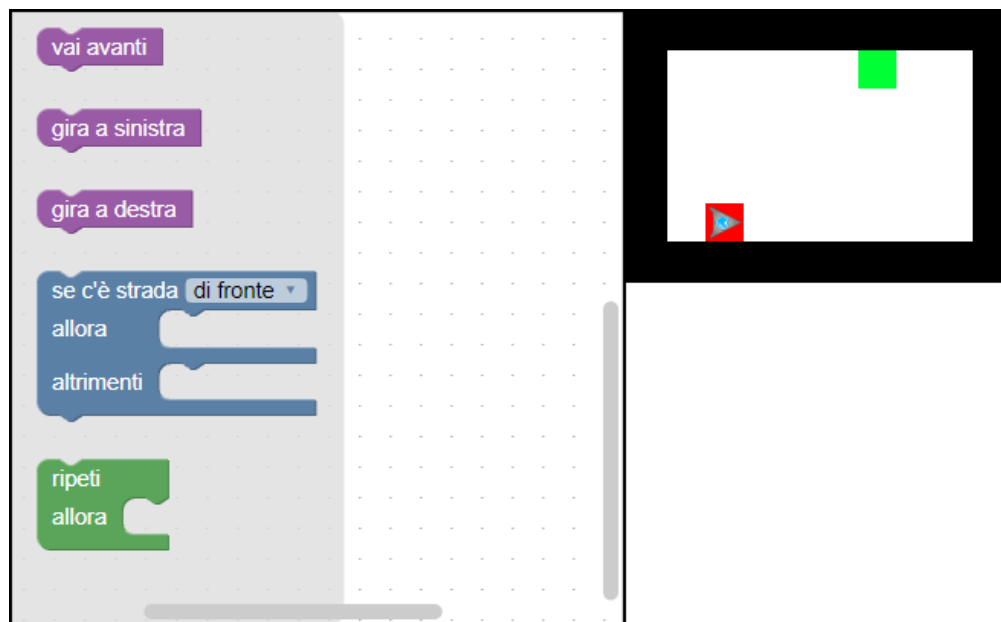
sistemi di riscrittura, semi-sistema di Thue, cifrario monoalfabetico

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



3. Un piccolo programma

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

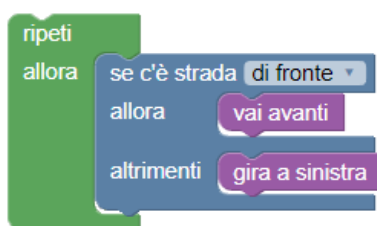


Trascina le istruzioni nel pannello di programmazione e uniscile nella giusta sequenza per permettere al robot di raggiungere il traguardo. Puoi usare solo un massimo di 4 istruzioni!



Soluzione

Una possibile soluzione è:



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

Questa è l'informatica!

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

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

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

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

Parole chiave e siti web

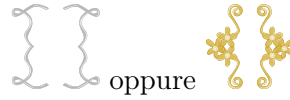
programmazione, cicli (loop), istruzioni condizionali, istruzioni

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

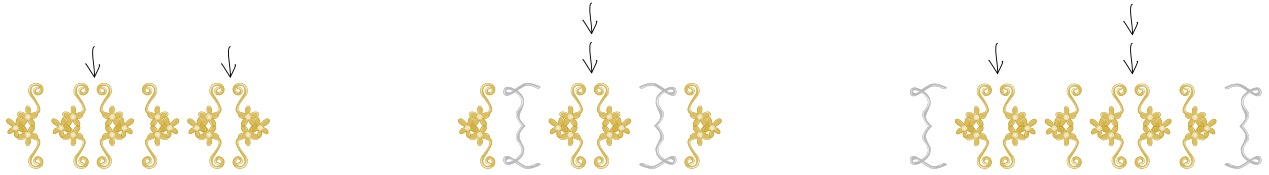


4. Gioielli

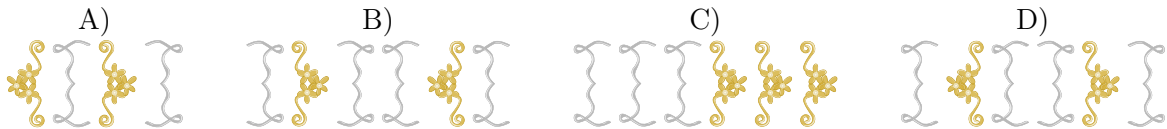
La famiglia Castoro produce artigianalmente dei gioielli in stile medioevale. Per fare questo, utilizza come base degli ornamenti a forma di parentesi, impiegati sempre a coppie. Per creare un gioiello, inizia da una delle seguenti coppie “di parentesi”:



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



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





Soluzione

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

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

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

Questa è l'informatica!

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

Parole chiave e siti web

espressione "ben formata", sintassi, semantica

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



5. Il giornale della scuola

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

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

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

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

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



Soluzione

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

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

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

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

Questa è l'informatica!

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

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

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

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

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

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

Parole chiave e siti web

pianificazione, grafo d'intervallo



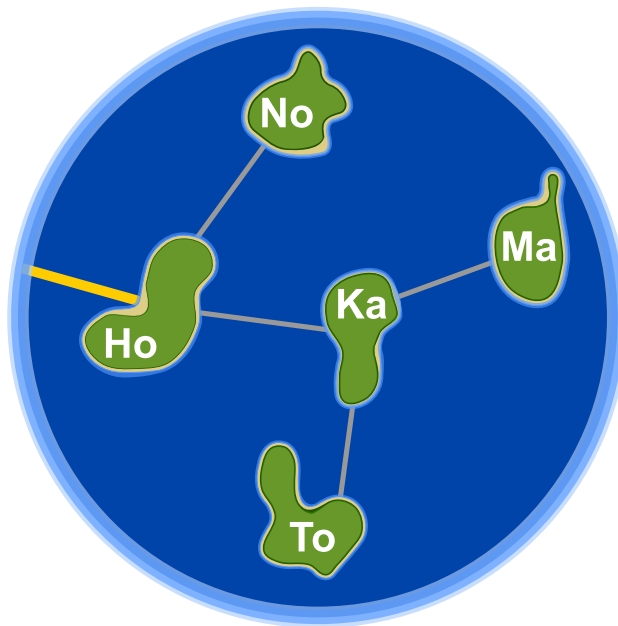
- https://it.wikipedia.org/wiki/Grafo_d'intervallo





6. Honomakato

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



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

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

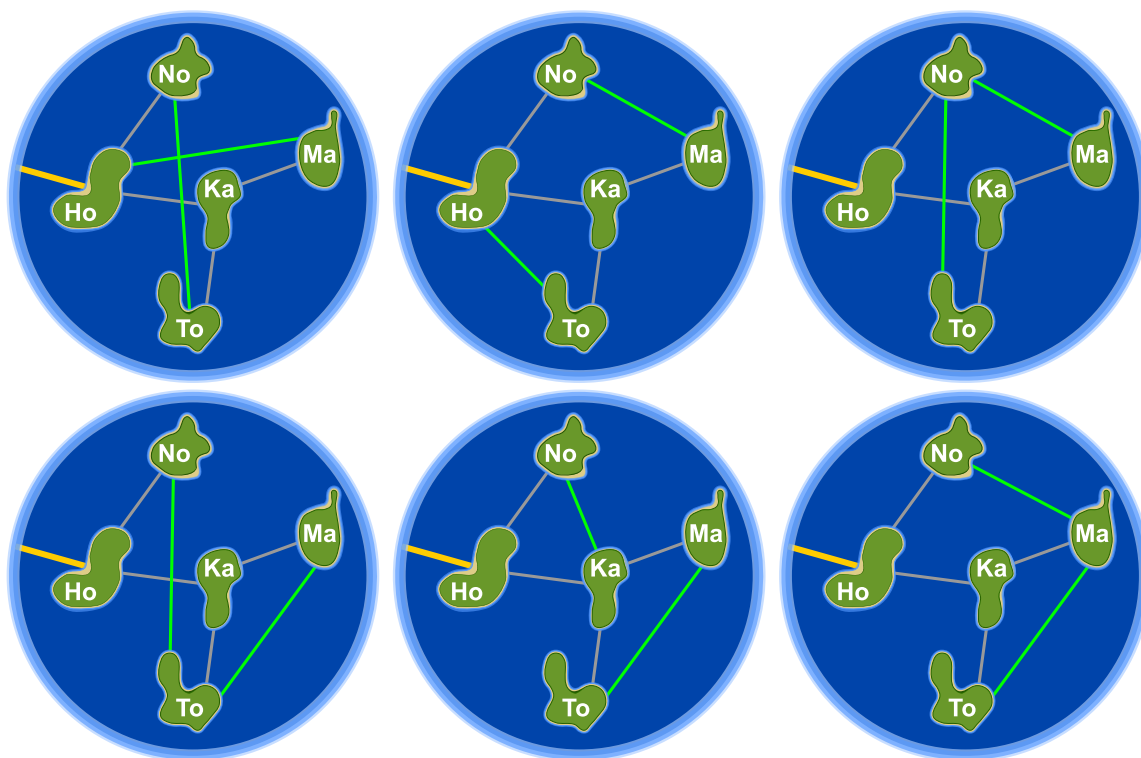


Soluzione

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

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

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



Questa è l'informatica!

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

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



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

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

Parole chiave e siti web

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

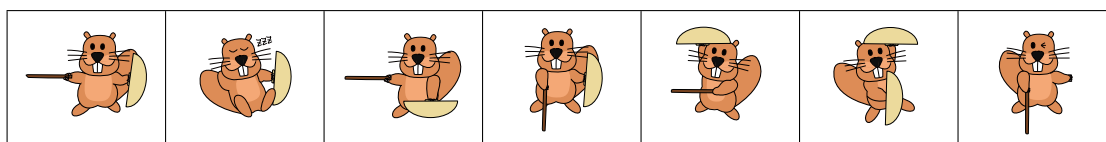
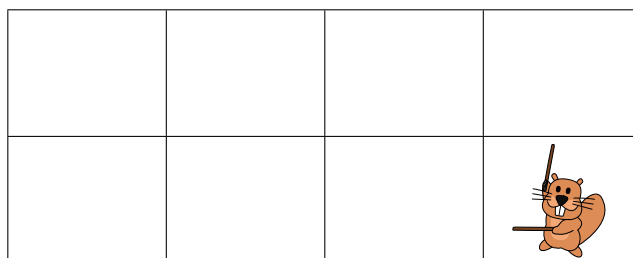
- https://en.wikipedia.org/wiki/Vertex_separator





7. L'arte del bastone giapponese

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

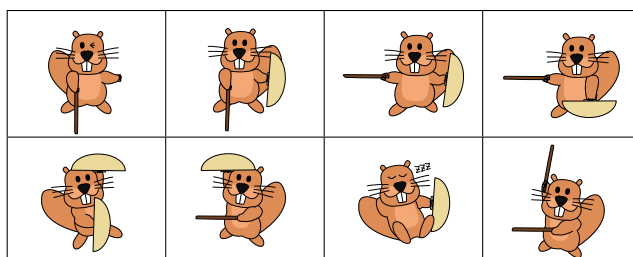


Associa le immagini degli amici di Lucia ai campi del cortile in modo che nella foto ogni bastone punti verso uno scudo.



Soluzione

La risposta corretta è:



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

Questa è l'informatica!

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

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

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

Parole chiave e siti web

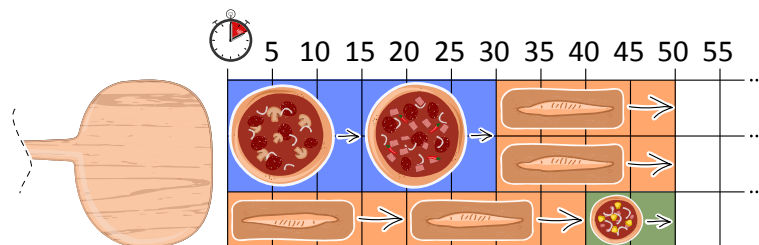
pensiero logico, conclusioni logiche

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



Soluzione

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



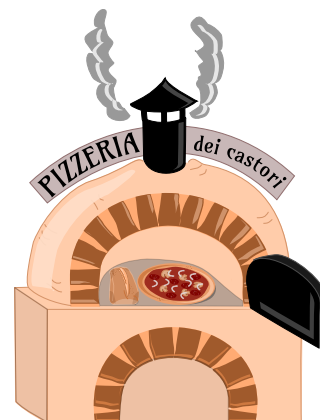
Queste soluzioni necessitano in totale di 50 minuti. Una soluzione più veloce non esiste, dato che il forno viene sempre utilizzato al massimo della sua capacità.

Questa è l'informatica!

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

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

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



Parole chiave e siti web

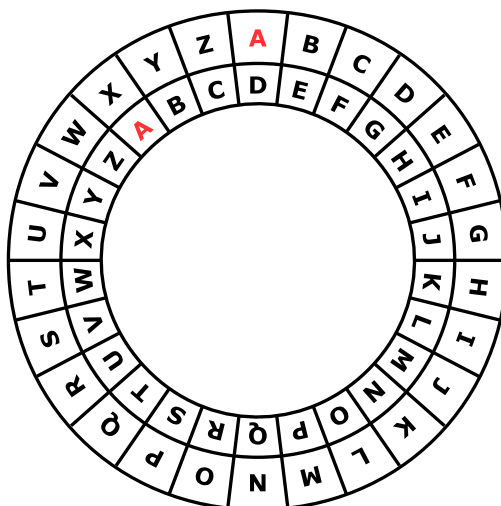
scheduling, liste di attesa, risorse, round-robin, algoritmo greedy

- <https://it.wikipedia.org/wiki/Scheduler>
- https://en.wikipedia.org/wiki/Round-robin_scheduling
- https://it.wikipedia.org/wiki/Algoritmo_greedy



9. Ordinazione segreta

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



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

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

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

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

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

Cosa ha ordinato Anna?

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



Soluzione

La risposta corretta è LASAGNA:

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

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

Questa è l'informatica!

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

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



Parole chiave e siti web

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

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



10. Gioco di monete

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



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

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



Soluzione

Non è possibile.

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

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

Questa è l'informatica!

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





Parole chiave e siti web








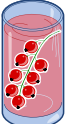



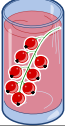



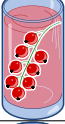
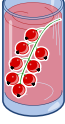



parità, bit di parità

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



11. 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

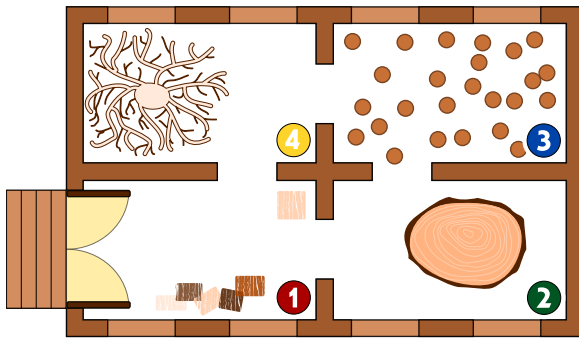
Ottimizzazione, accoppiamento (matching)

- [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))





12. Intrusione al museo



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

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

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

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

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

In quale minuto suonerà il sistema d'allarme?

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



Soluzione

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

Negli altri orari non è stato individuato alcun intruso:

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

Questa è l'informatica!

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

Parole chiave e siti web

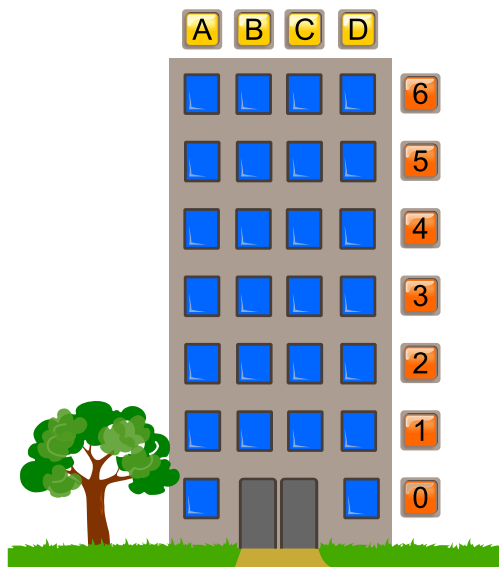
Sistema di sicurezza intelligente, regola, transizione di stato

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

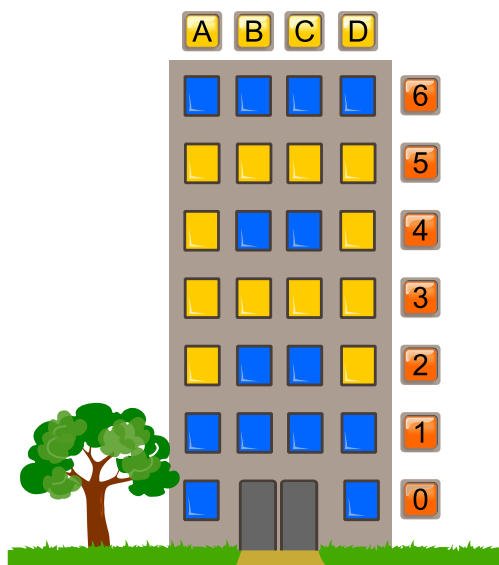


13. Giochi di luce

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 accendere o spegnere la luce di un intero piano o un'intera colonna di finestre.



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





Soluzione

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

Questa è l'informatica!

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

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

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

Parole chiave e siti web

Programmazione del hardware, sequenze di operazioni, operazioni binarie

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



14. 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

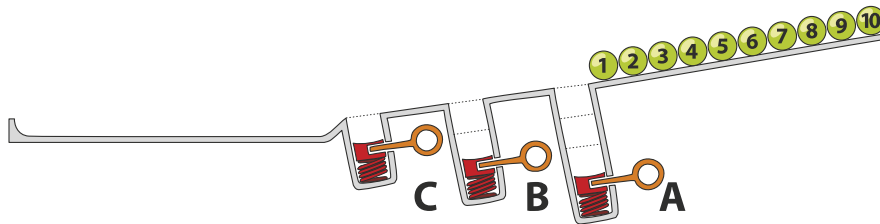
- https://it.wikipedia.org/wiki/Grammatica_formale



15. 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?

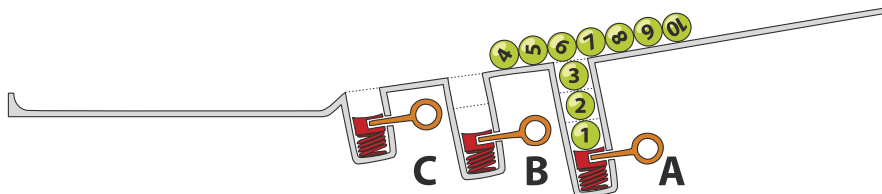
- A) B) C) D)



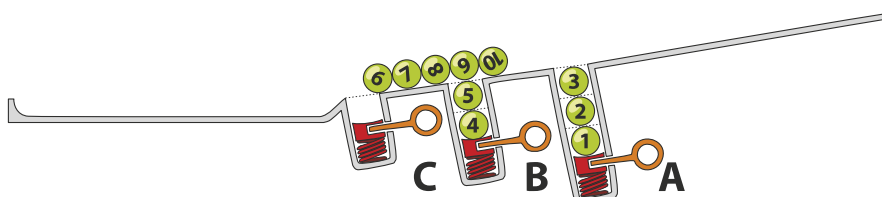
Soluzione

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

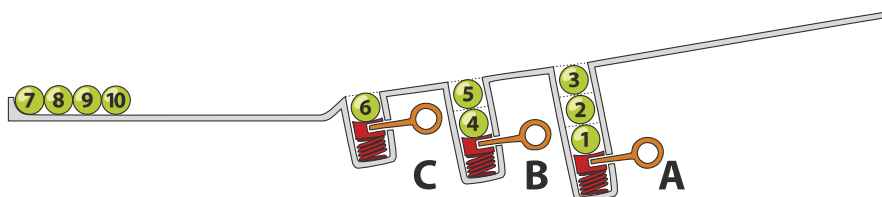
Le biglie 1, 2 e 3 cadono nella buca A.



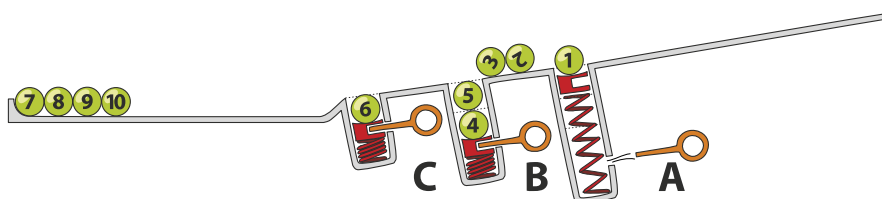
Le biglie 4 e 5 superano la buca A (ormai piena) e cadono nella buca B.



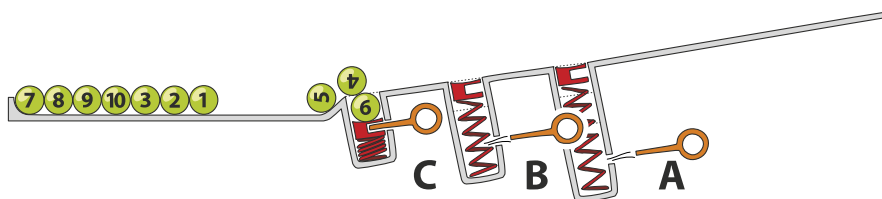
La biglia 6 cade nella buca C. Le biglie da 7 a 10 giungono invece alla fine della rampa.



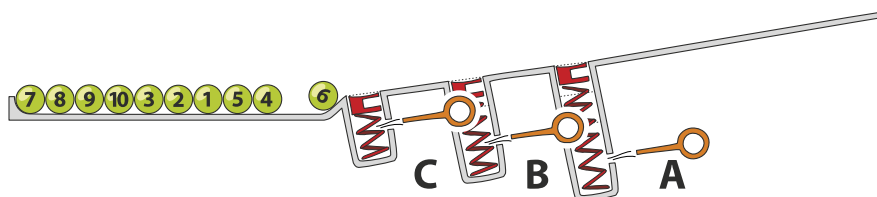
Dopo il rilascio della molla nella buca A, le biglie 3, 2 e 1 arrivano alla fine della rampa in ordine invertito rispetto alla partenza. La sequenza temporanea è quindi 7, 8, 9, 10, 3, 2, 1.



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))



A. Autori dei quesiti

 Andrea Adamoli
 Wilfried Baumann
 Daphne Blokhuis
 Eugenio Bravo
 Andrej Brodnik
 Carmen Bruni
 Amaury A. Castro Jr.
 Anton Chukhnov
 Raluca Constantinescu
 Valentina Dagiené
 Christian Datzko
 Susanne Datzko
 Janez Demšar
 Olivier Ens
 Hanspeter Erni

 Gerald Futschek
 Martin Guggisberg
 Urs Hauser
 Hans-Werner Hein
 Juraj Hromkovič
 Ungyeol Jung
 Ivana Kosírová
 Regula Lacher
 Greg Lee
 Milan Lukić
 Dario Malchiodi
 Hiroki Manabe
 Mattia Monga
 Zsuzsa Pluhár
 Wolfgang Pohl

 J.P. Pretti
 Daniel Rakijašić
 Chris Roffey
 Frances Rosamond
 Kirsten Schlüter
 Eljakim Schrijvers
 Maiko Shimabuku
 Taras Shpot
 Martin Stangl
 Gergely Tassy
 Ahto Truu
 Corina Elena Vint
 Michael Weigend
 Hongjin Yeh
 Momo Yokoyama



B. Sponsoring: concorso 2017

HASLERSTIFTUNG

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

ROBOROBO

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

d digitec.ch

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

bischofberger

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

verkehrshaus.ch

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

Museo Svizzero dei Trasporti

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

Standortförderung beim Amt für Wirtschaft und Arbeit
Kanton Zürich

 Information plus Automatik! Chunsch druus? Das ergibt Informatik.

i-factory (Museo Svizzero dei Trasporti, Lucerna)

 **UBS**

<http://www.ubs.com/>

Wealth Management IT and UBS Switzerland IT

bbv
Software Services

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

PRESENTEX
Das Geschenk - die gute Werbung

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



PH LUZERN
PÄDAGOGISCHE
HOCHSCHULE

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

ABZ

AUSBILDUNGS- UND BERATUNGSZENTRUM
FÜR INFORMATIKUNTERRICHT

<http://www.abz.inf.ethz.ch/>
Ausbildungs- und Beratungszentrum für Informatikunterricht 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
Informatik

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

senarclens
leu+partner
strategische kommunikation

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



C. Ulteriori offerte

010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001

SS!

www.svia-ssie-ssii.ch
schweizerischervereinfürinformatikind
erausbildung//société suisse pour l'infor
matique dans l'enseignement//società sviz
zera per l'informatica nell'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.