



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

Quesiti e soluzioni 2016 9^o e 10^o anno scolastico

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

A cura di:

Andrea Adamoli, Christian Datzko, Hanspeter Erni

010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
010010010100100100100001

SS!I

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



Hanno collaborato al Castoro Informatico 2016

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

Un particolare ringraziamento va a:

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

Valentina Dagiene: Bebras.org

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

Anna Morpurgo, Violetta Lonati, Mattia Monga: Italia

Gerald Futschek: Austrian Computer Society, Austria

Zsuzsa Pluhár: ELTE Informatikai Kar, Ungheria

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

Roman Hartmann: hartmannGestaltung (Flyer Castoro Informatico Svizzera)

Christoph Frei: Chragokyberneticks (Logo Castoro Informatico Svizzera)

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

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

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

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



INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Il Castoro Informatico 2016 è stato organizzato dalla Società Svizzera per l'Informatica nell'Insegnamento SSII. Il Castoro Informatico è un progetto della SSII con il prezioso sostegno della fondazione Hasler.

HASLERSTIFTUNG

Nota: Tutti i link sono stati verificati l'01.11.2016. Questo quaderno è stato creato il 13 novembre 2016 col sistema per la preparazione di testi L^AT_EX.



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



Premessa

Il concorso del «Castoro Informatico», presente già da diversi anni in molti paesi europei, ha l'obiettivo di destare l'interesse per l'informatica nei bambini e nei ragazzi. In Svizzera il concorso è organizzato in tedesco, francese e italiano dalla Società Svizzera per l'Informatica nell'Insegnamento (SSII), con il sostegno della fondazione Hasler nell'ambito del programma di promozione «FIT in IT».

Il Castoro Informatico è il partner svizzero del Concorso «Bebras International Contest on Informatics and Computer Fluency» (<http://www.bebas.org/>), situato in Lituania.

Il concorso si è tenuto per la prima volta in Svizzera nel 2010. Nel 2012 l'offerta è stata ampliata con la categoria del «Piccolo Castoro» (3^o e 4^o anno scolastico).

Il «Castoro Informatico» incoraggia gli alunni ad approfondire la conoscenza dell'Informatica: esso vuole destare interesse per la materia e contribuire a eliminare le paure che sorgono nei suoi confronti. Il concorso non richiede alcuna conoscenza informatica pregressa, se non la capacità di «navigare» in Internet poiché viene svolto online. Per rispondere alle domande sono necessari sia un pensiero logico e strutturato che la fantasia. I quesiti sono pensati in modo da incoraggiare l'utilizzo dell'informatica anche al di fuori del concorso.

Nel 2016 il Castoro Informatico della Svizzera è stato proposto a cinque differenti categorie d'età, suddivise in base all'anno scolastico:

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

Gli alunni iscritti al 3^o e 4^o anno scolastico hanno dovuto risolvere 9 quesiti (3 facili, 3 medi e 3 difficili).

A ogni altra categoria d'età sono stati assegnati 15 quesiti da risolvere, suddivisi in gruppi di cinque in base a tre livelli di difficoltà: facile, medio e difficile. Per ogni risposta corretta sono stati assegnati dei punti, mentre per ogni risposta sbagliata sono stati detratti. In caso di mancata risposta il punteggio è rimasto inalterato. Il numero di punti assegnati o detratti dipende dal grado di difficoltà del quesito:

	Facile	Medio	Difficile
Risposta corretta	6 punti	9 punti	12 punti
Risposta sbagliata	-2 punti	-3 punti	-4 punti

Il sistema internazionale utilizzato per l'assegnazione dei punti limita l'eventualità che il partecipante possa indovinare la risposta corretta.

Ogni partecipante aveva un punteggio iniziale di 45 punti (Piccolo Castoro 27).

Il punteggio massimo totalizzabile era pari a 180 punti (Piccolo castoro 108), mentre quello minimo era di 0 punti.

In molti quesiti le risposte possibili sono state distribuite sullo schermo con una sequenza casuale. Lo stesso quesito è stato proposto in più categorie d'età.



Per ulteriori informazioni:

SVIA-SSIE-SSII Società Svizzera per l'Informatica nell'Insegnamento

Castoro Informatico

Andrea Adamoli

castoro@castoro-informatico.ch

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

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



Indice

Hanno collaborato al Castoro Informatico 2016	i
Premessa	ii
1. Messaggi segreti	1
2. Gerarchia	3
3. Costruzione di ponti	5
4. Carte nascoste	7
5. Lavoro di gruppo	9
6. Bomboniere	13
7. Attraverso il tunnel	15
8. Stretta di mano	17
9. Gioco del Nim	19
10. Numeri di maglia ordinati	21
11. Montacarichi	23
12. Gioco con le biglie	27
13. Due possibilità	31
14. Incrocio di tram	33
15. La codifica delle bandiere	35
A. Autori dei quesiti	37
B. Sponsoring: concorso 2016	38
C. Ulteriori offerte	40



1. Messaggi segreti

Gli agenti Boris e Berta comunicano attraverso messaggi segreti che nessuno dovrebbe capire all'infuori di loro stessi. Boris desidera inviare a Berta il seguente messaggio:

RITROVOCONBENALLE6

Egli scrive i singoli caratteri del testo nelle celle di una tabella di quattro colonne per cinque righe, procedendo da sinistra a destra e dall'alto verso il basso. Se alla fine restano delle celle vuote, aggiunge degli asterischi (*). L'immagine mostra il risultato di tale operazione:

R	I	T	R
O	V	O	C
O	N	B	E
N	A	L	L
E	6	*	*

Boris genera poi il messaggio segreto nel modo seguente: su un nuovo foglio, riscrive le lettere questa volta seguendo la tabella dall'alto in basso, colonna per colonna da sinistra a destra.

ROONEIVNA6TOBL*RCEL*

Berta utilizza lo stesso metodo per scrivere la risposta segreta a Boris:

SEEAETNNNIAEGCOBVOH*

Cosa ha risposto Berta?

- A) STABENEPOSSOVENIRE
- B) STABENEVENGOANCHEIO
- C) SEVUOIVENGOANCHEIO
- D) VORREIVENIREPUREIO



Soluzione

Il testo iniziale si trova in questo modo: si scrive il messaggio ricevuto nella tabella di quattro colonne per cinque righe, questa volta dall'alto in basso, colonna per colonna a partire da sinistra.

S	T	A	B
E	N	E	V
E	N	G	O
A	N	C	H
E	I	O	*

Si legge poi il messaggio decifrato riga per riga, ottenendo il testo seguente:

STABENEVENGOANCHEIO

L'asterisco alla fine non appartiene al messaggio.

Questa è l'informatica!

I messaggi inviati tramite computer possono essere facilmente intercettati. Quando questi messaggi contengono dati sensibili, come password o informazioni personali, desideriamo che solo il destinatario e nessun altro possa leggere il testo. In questi casi si può cifrare il messaggio in chiaro trasformandolo in un messaggio segreto. Solo il destinatario saprà come decifrarlo per ottenerne il testo originale.

Nella pratica esistono molti metodi per cifrare messaggi. Il metodo applicato in questo esercizio è chiamato trasposizione ed è applicato da ca. 2400 anni. Con esso tutte le lettere del messaggio originale rimangono invariate, ciò che cambia è la loro posizione. Questo metodo è molto semplice da violare, di fatto non si tratta di vera e propria cifratura, ma piuttosto di occultamento dell'informazione.

La crittografia è la scienza che si occupa di studiare metodi per la cifratura e rappresenta un ramo importante dell'informatica. Imprese commerciali e banche presenti su Internet necessitano metodi sicuri di cifratura. Al giorno d'oggi le cifrature sono eseguite dai computer e si basano su metodi matematici, che rendono praticamente impossibile violare i messaggi, se non ne si conosce la chiave.

Siti web e parole chiave

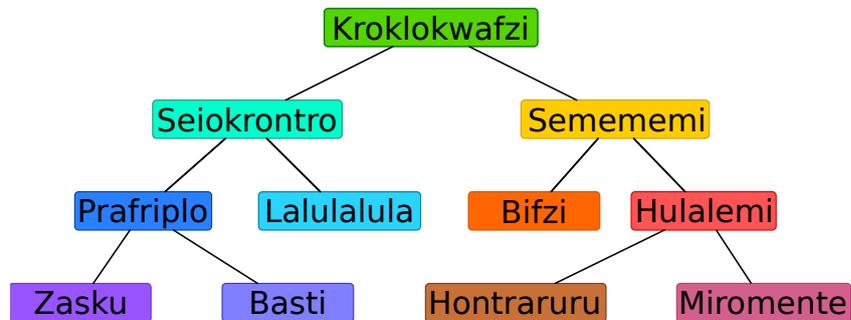
crittografia, cifratura, trasposizione, protezione dei dati

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



2. Gerarchia

L'immagine rappresenta le relazioni tra specie animali del pianeta Venere. Una linea che collega due specie indica che tutti gli animali appartenenti alla specie della casella posta più in basso sono anche animali della specie indicata nella casella più in alto.



Per esempio, gli animali della specie “Hulalemi” sono anche della specie “Semememi”. D’altro canto, qualche “Seiokronto” non fa parte dei “Basti”.

Solo una delle seguenti affermazioni è vera, quale?

- A) Tutti i Basti sono anche Seiokronto.
- B) Qualche Hontraruru non è anche Semememi.
- C) Tutti i Zasku sono anche Bifzi.
- D) Tutti i Prafriplo sono anche Basti.



Soluzione

La risposta corretta è A).

- A) Tutti gli animali della specie Basti sono anche della specie Prafriplo. Dato che gli animali della specie Prafriplo sono Seiokrontro, ne consegue che tutti i Basti sono anche della specie Seiokrontro.
- B) Gli Hontraruru sono animali della specie Hulalemi, i quali a loro volta sono dei Semememi. Quindi tutti i Hontraruru sono della specie Semememi.
- C) I Zasku sono anche della specie Prafriplo e non hanno alcuna relazione con la specie dei Bifzi.
- D) Tutti i Basti sono anche della specie Prafriplo, ma non vale il contrario.

Questa è l'informatica!

Le relazioni tra le diverse specie di animali sono rappresentate attraverso uno schema che in informatica è detto "albero". I biologi utilizzano l'albero "filogenetico" per illustrare le relazioni fra le diverse specie.

In informatica si utilizzano spesso degli alberi per rappresentare le relazioni graficamente. Un albero genealogico rappresenta ad esempio le relazioni tra figli, genitori e nonni. Quando si utilizza questa rappresentazione grafica le relazioni tra i vari "nodi" possono essere immediatamente comprese.

Gli alberi sono anche un eccellente mezzo per memorizzare dei dati in modo ordinato e ritrovarli poi velocemente: con pochi passi è possibile accedere in un lampo a enormi insiemi di informazioni.

Siti web e parole chiave

alberi, specializzazione, generalizzazione

- [https://it.wikipedia.org/wiki/Albero_\(grafo\)](https://it.wikipedia.org/wiki/Albero_(grafo))
- https://it.wikipedia.org/wiki/Albero_filogenetico



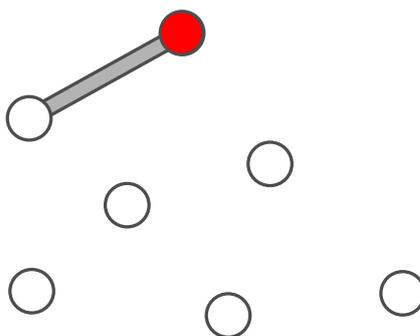
3. Costruzione di ponti

Nonno castoro è diventato timoroso nei confronti dell'acqua. Egli vorrebbe quindi andare dalla propria tana a quella dei familiari attraverso dei ponti. Gli altri castori vogliono molto bene al nonno e per questo desiderano costruire dei ponti che osservino le seguenti regole:

- Il nonno deve poter andare dalla propria tana a qualsiasi altra, attraversando non più di due ponti.
- Oltre al ponte che conduce a una certa tana, possono essercene al massimo altri due che ne escono.

I castori iniziano ad elaborare il progetto. Su un foglio disegnano tutte le loro tane, rappresentandole attraverso dei cerchi. La tana del nonno è un cerchio colorato di rosso. Con una linea pianificano il primo ponte in uscita dalla tana del nonno ... ma poi non sanno più come continuare.

Completa il piano in modo da osservare tutte le regole descritte sopra. Esistono più soluzioni, ma in ogni caso devono essere disegnati altri 5 ponti.

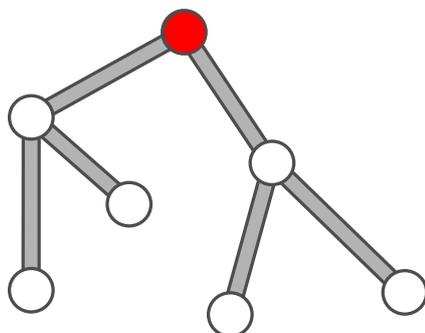




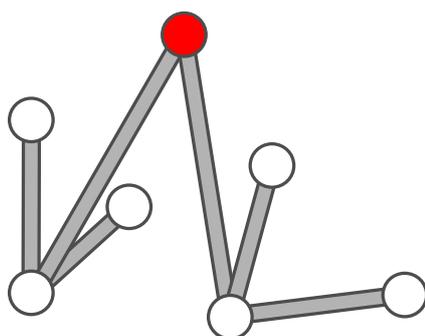
Soluzione

Il progetto esposto qui di seguito soddisfa le regole, poiché:

- dal cerchio rosso si raggiunge ogni altro cerchio attraversando non più di due linee;
- da ogni cerchio escono al massimo 3 linee.



Esistono comunque anche altre soluzioni, ad esempio:



Questa è l'informatica!

Nonostante esistano molte soluzioni, tutte hanno lo stesso schema generale. Dalla tana rossa escono due ponti che si collegano ad altre due tane e, da ognuna di queste, altri due ponti ancora che uniscono le restanti quattro tane. Se ci fosse un'ulteriore tana, il progetto non potrebbe essere realizzato rispettando le regole.

I castori costruiscono quindi un cosiddetto *albero*: tutti i *nodi* (cerchi / tane) possono essere raggiunti percorrendo gli *archi* (linee / ponti). Le regole ci obbligano a utilizzare un tipo particolare di albero: il fatto che solo due archi possono uscire da un nodo fa sì che l'albero sia *binario*; il fatto che dobbiamo percorrere al massimo 2 archi per visitare un nodo qualsiasi fa sì che esso sia *minimale*.

Siti web e parole chiave

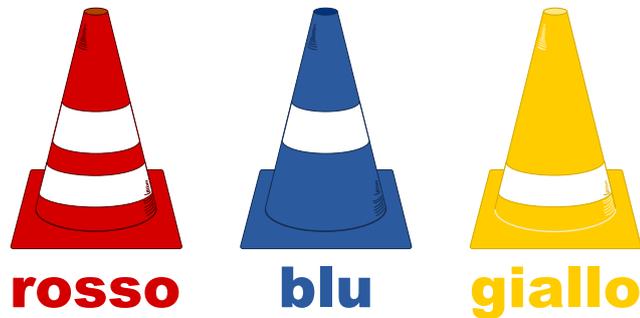
Albero

- [https://it.wikipedia.org/wiki/Albero_\(grafo\)](https://it.wikipedia.org/wiki/Albero_(grafo))
- https://it.wikipedia.org/wiki/Albero_binario



4. Carte nascoste

Vanessa nasconde delle carte sotto a tre coni stradali. Ogni nuova carta viene sempre inserita sotto a un determinato cono. Se il cono nasconde già una carta, quella nuova viene inserita sopra ad essa.



Per descrivere quale carta è stata inserita sotto a quale cono, Vanessa usa la seguente notazione:

rosso † 5

rosso † 3



Vanessa ha annotato:

rosso † 3

giallo † 5

rosso † 6

giallo † 8

blu † 1

giallo † 3

Alla fine, quale carta è posta in cima al mazzo nascosto sotto ad ogni cono?

- A) rosso: 3, blu: 1, giallo: 5
- B) rosso: 9, blu: 1, giallo: 16
- C) rosso: 6, blu: 1, giallo: 3
- D) rosso: 8, blu: 1, giallo: 3



Soluzione

All'inizio nessuna carta è stata nascosta sotto ai coni. La tabella seguente mostra quali carte sono inserite in cima, dopo ogni annotazione:

Annotazione	rosso	blu	giallo
	-	-	-
rosso \leftarrow 5	5	-	-
giallo \leftarrow 5	5	-	5
rosso \leftarrow 6	6	-	5
giallo \leftarrow 8	6	-	8
blu \leftarrow 1	6	1	8
giallo \leftarrow 3	6	1	3

È quindi chiaro che C) è la risposta corretta.

Questa è l'informatica!

Il cono stradale può essere interpretato come una variabile, il cui nome è, rispettivamente, “rosso”, “blu” e “giallo”. Una variabile è in pratica uno spazio di memoria (“un cassetto”) del computer in cui poter salvare dei valori (nel nostro caso dei numeri interi). Ogni comando è di fatto un'assegnazione che sostituisce il valore salvato in precedenza con un nuovo valore.

Nei linguaggi di programmazione, le assegnazioni vengono spesso denotate con “=” o “:=”. Questo porta però spesso ad errore di concezione: mentre in matematica, l'espressione “ $x = x + 1$ ” è errata, in informatica significa che la variabile x incrementa il proprio valore di 1 (“il nuovo valore di x è pari al vecchio valore di x più 1”). Spesso, al di fuori dei linguaggi di programmazione, si utilizza la freccia “ \leftarrow ” per indicare un'assegnazione, poiché essa è meglio compresa dai matematici e non scontenta gli informatici.

Siti web e parole chiave

variabile, assegnazione

- [https://it.wikipedia.org/wiki/Variabile_\(informatica\)](https://it.wikipedia.org/wiki/Variabile_(informatica))
- https://it.wikipedia.org/wiki/Operatore_di_assegnamento



5. Lavoro di gruppo

Sarah e i suoi amici Alicia, Beat, Caroline, David ed Emil devono svolgere un lavoro di gruppo in classe. Essi suddividono il lavoro in modo tale che Sarah abbia il compito di raggruppare i risultati dei compagni. Sarah può ricevere direttamente i risultati del lavoro di Emil, ma per ottenere quelli degli altri deve osservare le seguenti regole:

- Per ricevere i risultati da David, deve prima ottenere quelli di Alicia.
- Per ricevere i risultati da Beat, deve prima ottenere quelli di Emil.
- Per ricevere i risultati da Caroline, deve prima ottenere quelli di Beat e David.
- Per ricevere i risultati da Alicia, deve prima ottenere quelli di Beat ed Emil.

Trascina in nomi nella giusta sequenza, in modo che Sarah possa ricevere tutti i risultati:





Soluzione

La risposta corretta è:

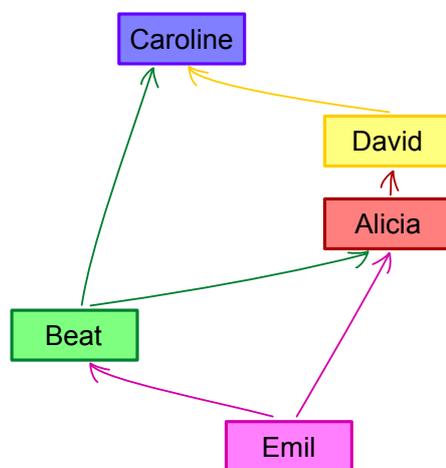


Emil è l'unico a poter consegnare i risultati direttamente, senza aspettare alcun altro compagno. Per questo deve essere posto in cima alla lista. Beat è l'unico compagno che dipende solo da Emil per svolgere il proprio lavoro. Con i risultati di Emil e Beat, Alicia è l'unica a poter svolgere il proprio compito. Analogamente, David è l'unico a poter continuare dopo aver ottenuto i risultati di Alicia e infine anche Caroline può finalmente svolgere il proprio compito.

Questa è l'informatica!

Capita spesso nella vita che alcune cose possano essere portate a termine solo se determinati prerequisiti sono soddisfatti. Anche in informatica ci si occupa di simili operazioni di pianificazione (scheduling). I programmi informatici sviluppati per l'industria hanno spesso a che fare con questi problemi: alcuni processi complicati, come ad esempio la costruzione di automobili, devono essere molto efficienti ed evitare che la catena di montaggio si blocchi per attendere dei pezzi. In questo modo si possono risparmiare molti soldi, sia riducendo il bisogno di immagazzinare parti di auto che incrementando l'utilizzo delle macchine.

Per modellare una pianificazione, si utilizzano dei grafi i cui nodi sono le varie operazioni (nel nostro caso la raccolta dei risultati di un compagno) e gli archi orientati le relazioni di dipendenza. Un grafo del genere viene rappresentato in questo modo:



Ogni possibile soluzione per il problema è rappresentata da percorso attraverso il grafo che visiti tutti i nodi. Nel nostro caso c'è solo un percorso possibile: Emil → Beat → Alicia → David → Caroline.

Trovare una soluzione ad un problema di pianificazione non è però sempre semplice. Esistono delle situazioni complesse in cui, ad esempio, alcune operazioni possono essere svolte in parallelo. In questi casi è anche spesso necessario ottimizzare i costi in termini di tempo o soldi. Di frequente risulta impraticabile individuare e verificare ogni soluzione possibile, per questo vengono utilizzati algoritmi euristici (approssimativi) per individuare soluzioni ottimali.



Siti web e parole chiave

scheduling, pianificazione, grafo orientato

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



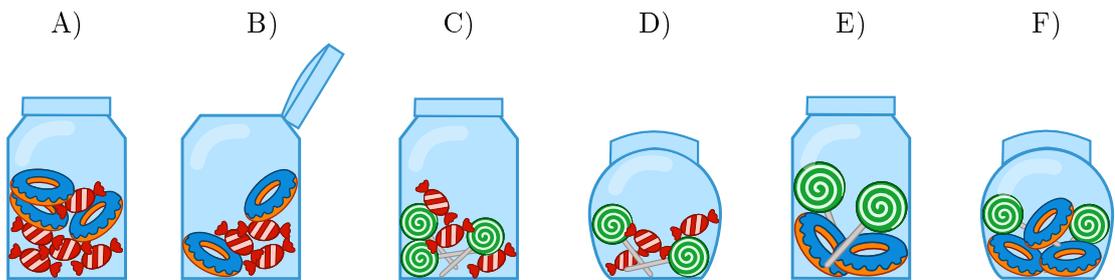


6. Bomboniere

La bomboniera è un contenitore (anche di vetro) per dolci. Carl e Judy hanno 3 bomboniere ciascuno: Carl possiede le bomboniere A), B) e C), mentre Judy possiede le bomboniere D), E) e F). Ogni bomboniera ha le seguenti cinque caratteristiche:

- La bomboniera può essere aperta o chiusa.
- La bomboniera può contenere caramelle a strisce rosse e bianche oppure no.
- La bomboniera può contenere ciambelle blu oppure no.
- La bomboniera può contenere lecca-lecca a spirale verde oppure no.
- La bomboniera può essere tondeggiante oppure angolosa (spigolosa).

Indica la bomboniera che possiede sia le caratteristiche in comune a tutte le bomboniere di Carl che quelle in comune a tutte le bomboniere di Judy.





Soluzione

Le ciambelle blu non devono essere considerate, poiché non sono una caratteristica in comune a tutte le bomboniere di Carl né a quelle di Judy. Esse sono infatti presenti in tutte le bomboniere tranne che in C) e D).

Le due caratteristiche in comune alle bomboniere di Carl A), B) e C) sono:

- ogni bomboniera è angolosa;
- ogni bomboniera contiene caramelle a strisce rosse e bianche.

Le due caratteristiche in comune alle bomboniere di Judy D), E) e F) sono:

- ogni bomboniera è chiusa;
- ogni bomboniera contiene lecca-lecca a spirale verde.

Ora, solo la bomboniera C) possiede tutte le quattro caratteristiche: è angolosa, contiene caramelle a strisce rosse e bianche, contiene lecca-lecca a spirale verde ed è chiusa.

Questa è l'informatica!

Nei propri modelli di dati, l'informatica predilige raggruppare gli oggetti con caratteristiche (meglio, "proprietà" o "attributi") comuni. Nel nostro quesito abbiamo a che fare con 5 proprietà e 2 due gruppi. Il compito consiste nel cercare oggetti che posseggano tutte le proprietà che i due gruppi hanno rispettivamente in comune. Nelle basi di dati relazionali (e in matematica), questa operazione viene detta "intersezione di insiemi".

Nelle basi di dati relazionali, si possono identificare (o, meglio, *filtrare*) velocemente degli oggetti con determinate proprietà in insiemi enormi ("costruzione di sotto-insiemi"). Per esempio, nei negozi on-line è possibile filtrare gli smartphone con determinate proprietà per accumulatori, display o altri elementi ancora.

Nella costruzione delle basi di dati è importante predisporre in modo dettagliato quali proprietà utilizzare nel modello dei dati. Se importanti proprietà dovessero essere dimenticate, sarà difficile in futuro selezionare l'oggetto con le caratteristiche volute. Se, invece, dovessero essere contemplate delle proprietà di scarsa rilevanza, si renderebbe la base di dati inutilmente complessa e costosa.

Siti web e parole chiave

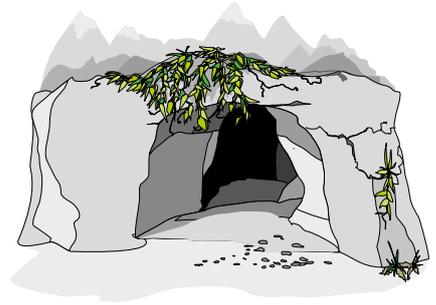
orientamento agli oggetti, attributi, logica

- [https://it.wikipedia.org/wiki/Intersezione_\(insiemistica\)](https://it.wikipedia.org/wiki/Intersezione_(insiemistica))
- [https://it.wikipedia.org/wiki/Unione_\(insiemistica\)](https://it.wikipedia.org/wiki/Unione_(insiemistica))
- https://it.wikipedia.org/wiki/Teoria_degli_insiemi
- https://it.wikipedia.org/wiki/Base_di_dati



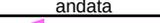
7. Attraverso il tunnel

Anna e Ben fanno una passeggiata con i loro genitori. Lungo il cammino c'è un tunnel. Per esperienza sanno che ognuno di essi impiega un tempo diverso a percorrerlo: Anna impiega 10 minuti, Ben 5 minuti, la madre 20 minuti e il padre 25 minuti. Attraverso lo stretto e scuro tunnel possono passare solo una o, al massimo, 2 persone. Essi devono compiere quindi molti passaggi. Quando si attraversa il tunnel in due, si impiega tanto tempo quanto la persona più lenta. Nel tunnel bisogna assolutamente utilizzare una torcia.



Quando la famiglia si trova all'entrata del tunnel, si accorge che l'accumulatore della torcia ha energia solo per altri 60 minuti. Rusciranno a passare tutti attraverso il tunnel prima che la torcia si spenga? Anna ne è certa: "Sicuro, possiamo farcela in 5 passaggi!"

Sposta i nomi nelle celle appropriate, in modo da attuare il piano di Anna.

 andata		
 ritorno		
 andata		
 ritorno		
 andata		

Anna

Ben

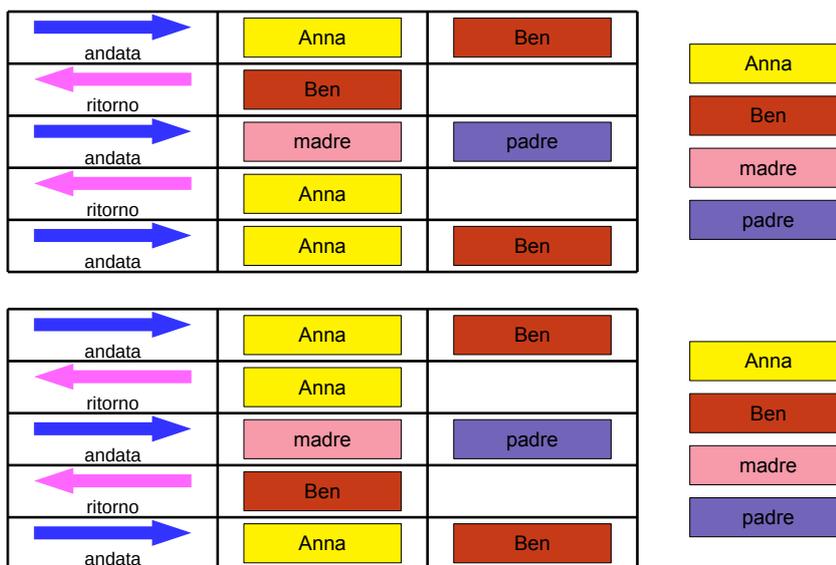
madre

padre



Soluzione

Passare tutti attraverso il tunnel in non più di 60 minuti è possibile solo se il padre e la madre (i più lenti) compiono il tragitto una sola volta. Questo è realizzabile se dapprima Anna e Ben lo attraversano e poi uno dei due torna indietro per portare la torcia ai genitori. Essi attraverseranno il tunnel assieme e il fratello rimanente porterà di nuovo indietro la torcia per scortare il primo all'uscita. Con questo metodo è indifferente se il primo a tornare indietro è Ben oppure Anna: entrambi dovranno tornare una volta, impiegando complessivamente lo stesso tempo.



Questa è l'informatica!

In informatica succede spesso che ci siano risorse limitate per risolvere un problema. Allo stesso tempo la soluzione trovata deve anche soddisfare delle condizioni accessorie. Il problema viene risolto delineando un piano che tiene conto dell'utilizzo delle risorse e si cura di soddisfare le condizioni accessorie.

In informatica problemi simili vengono risolti con algoritmi di *scheduling*. Dato che i problemi sono molto complessi, il computer impiega anche molti anni per calcolare una soluzione ottimale. Questa soluzione non deve essere necessariamente la migliore, basta che soddisfi tutte le condizioni. Anche nel nostro caso, il problema richiedeva il passaggio del tunnel entro 60 minuti e non il passaggio nel più breve tempo possibile. Il fatto che la soluzione proposta fosse comunque la migliore è solo un caso: l'importante, come detto, era trovarne una ottimale ... e poi passare al prossimo quesito.

Siti web e parole chiave

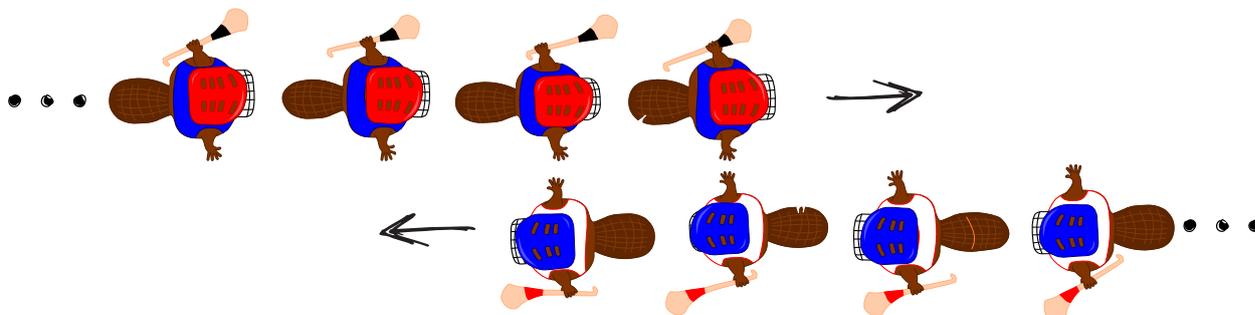
(job) scheduling, condizioni, risorse

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



8. Stretta di mano

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



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

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

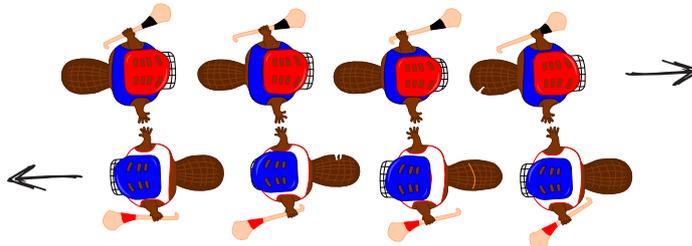
Quanti secondi si impiegano in totale per la stretta di mano tra tutti i membri delle due squadre?



Soluzione

La risposta corretta è 29.

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



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

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

Questa è l'informatica!

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

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

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

Siti web e parole chiave

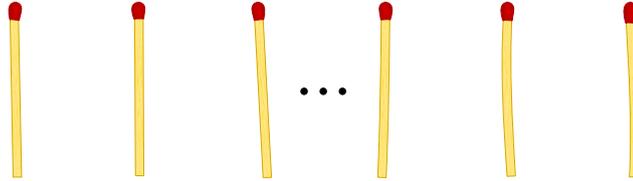
complessità temporale, analisi del tempo di esecuzione (run-time)

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



9. Gioco del Nim

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



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

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

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



Soluzione

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

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

Questa è l'informatica!

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

Siti web e parole chiave

gioco del Nim, teoria dei giochi, albero di decisione

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



10. Numeri di maglia ordinati

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

Squadra 1:



Squadra 2:



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

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



Soluzione

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

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

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

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

Questa è l'informatica!

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

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

Siti web e parole chiave

ordinamento, ricerca binaria (o dicotomica)

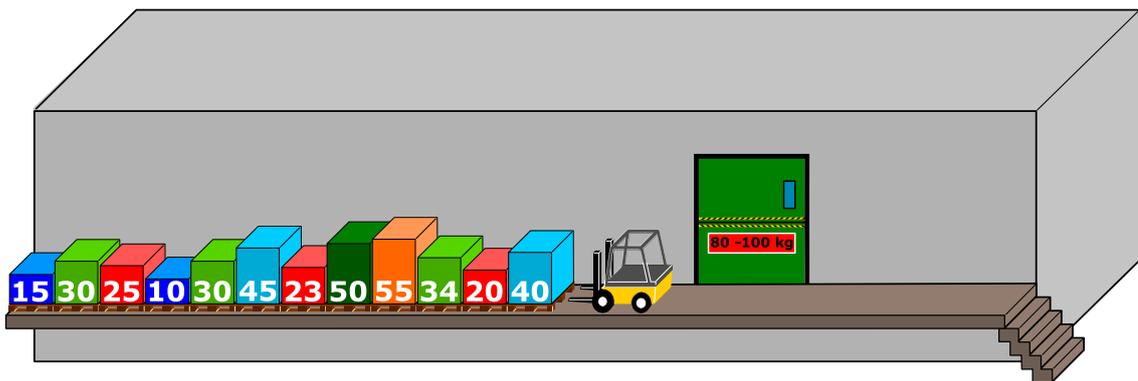
- https://it.wikipedia.org/wiki/Ricerca_dicotomica



11. Montacarichi

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

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



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

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

Quale affermazione è corretta tra quelle esposte qui di seguito?

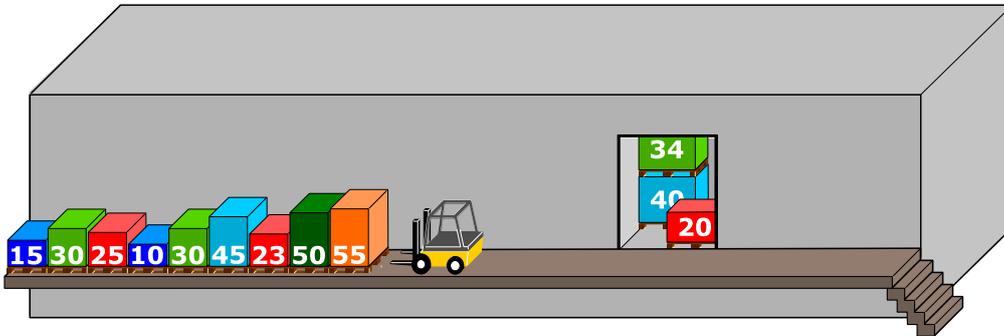
- A) Al secondo viaggio del montacarichi il peso complessivo ammontava a 98 kg.
- B) Non si è mai formata una fila a destra del montacarichi.
- C) In un viaggio del montacarichi il peso complessivo ammontava a 100 kg.
- D) Il montacarichi ha fatto in totale 5 viaggi.
- E) Non è possibile, con la procedura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



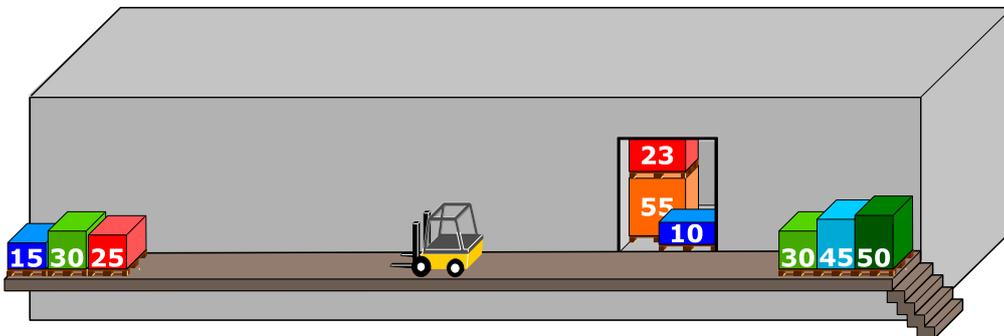
Soluzione

La risposta corretta è C):

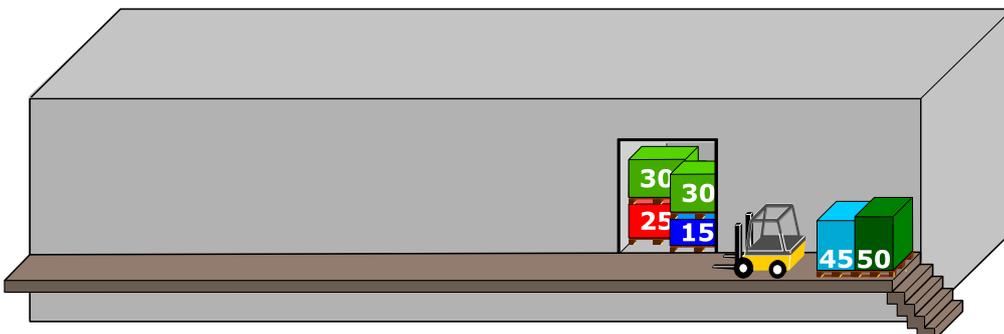
- Il primo carico è composto dai primi tre bancali. Questi pesano: $40\text{ kg} + 20\text{ kg} + 34\text{ kg} = 94\text{ kg}$.



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

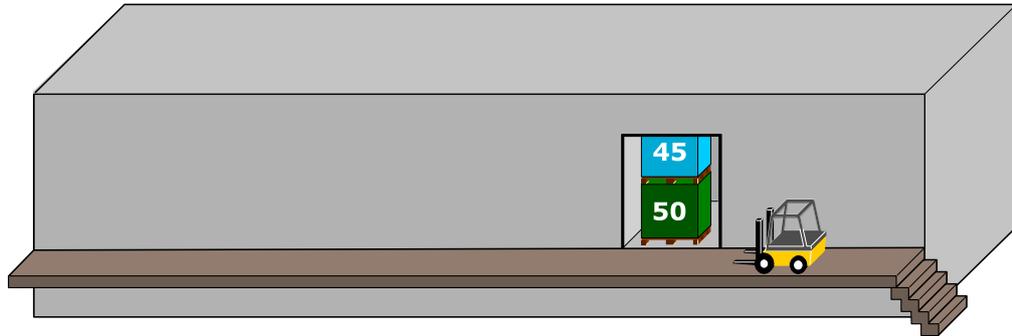


- Gli ultimi 3 bancali ($25\text{ kg} + 30\text{ kg} + 15\text{ kg} = 70\text{ kg}$) vengono caricati sul montacarichi. Dato che il peso è inferiore agli 80 kg, viene aggiunto anche il primo bancale della fila di destra (30 kg), raggiungendo così un totale di 100 kg.





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



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

Questa è l'informatica!

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

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

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

Siti web e parole chiave

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

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





12. Gioco con le biglie

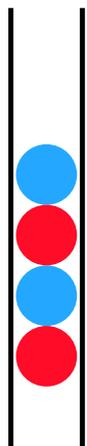
Emil gioca a un nuovo solitario sul suo computer. Il gioco inizia con un minimo di 3 biglie colorate (rosse o blu) impilate in un tubo.

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

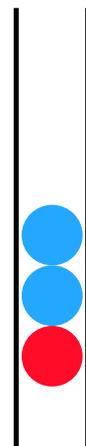
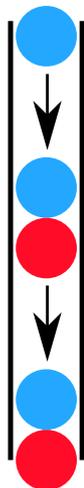
Se la biglia più in basso era rossa, una nuova biglia blu viene impilata in alto:

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

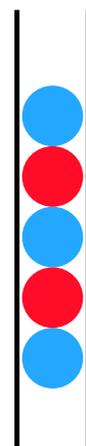
prima



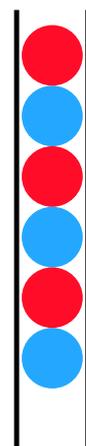
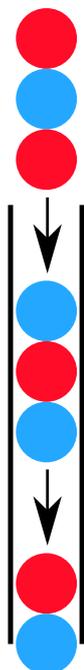
dopo



prima



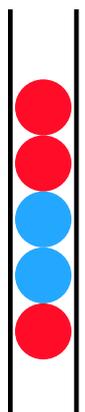
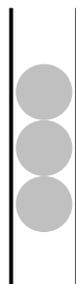
dopo



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

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

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

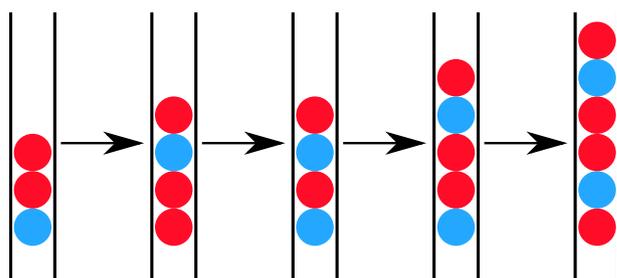
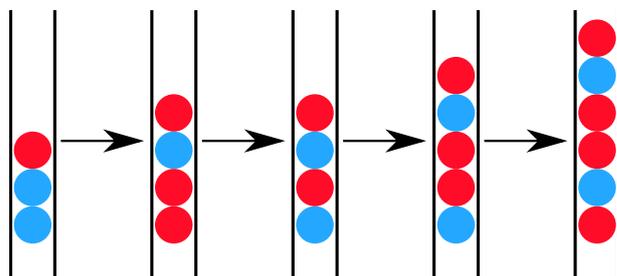
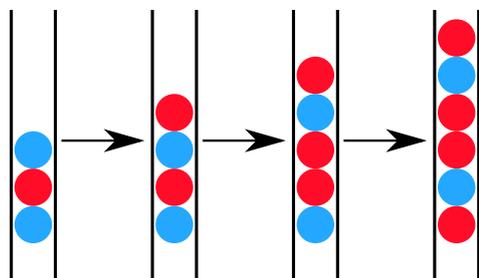
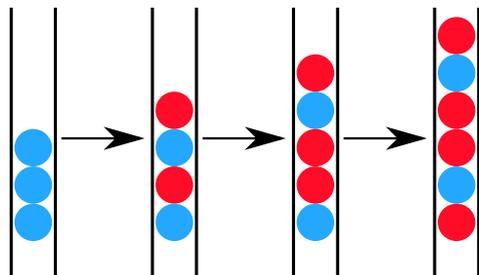




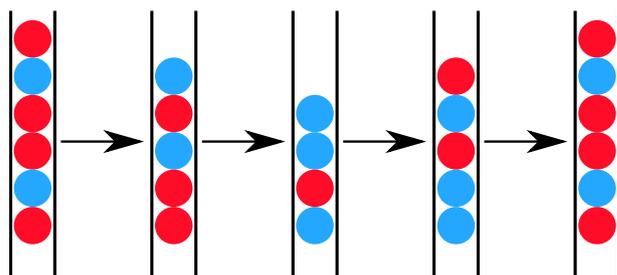
Soluzione

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

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



Con questa configurazione si entrerebbe poi in un ciclo infinito:





Questa è l'informatica!

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

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

$$Xbb \rightarrow rbrX$$

$$Xrb \rightarrow rbrX$$

$$Xbr \rightarrow bX$$

$$Xrr \rightarrow bX$$

Siti web e parole chiave

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

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



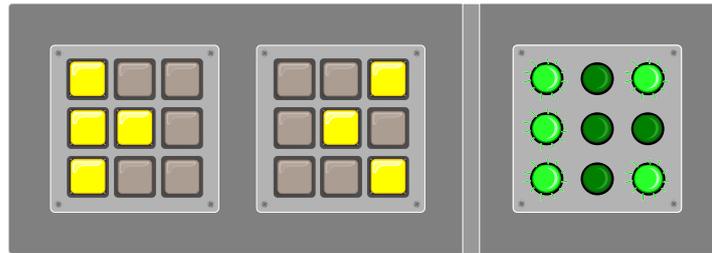


13. Due possibilità

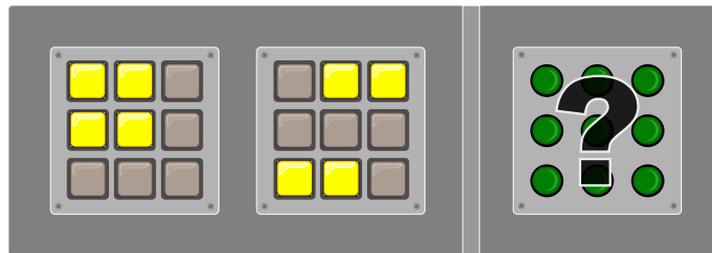
Erich ha trovato un vecchio apparecchio elettronico. Sulla parte sinistra ci sono 2 campi, ognuno dei quali ha 9 tasti. Sulla parte destra c'è invece un campo con 9 luci LED. Premendo i tasti dei campi a sinistra si accendono o si spengono i LED sulla destra.

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

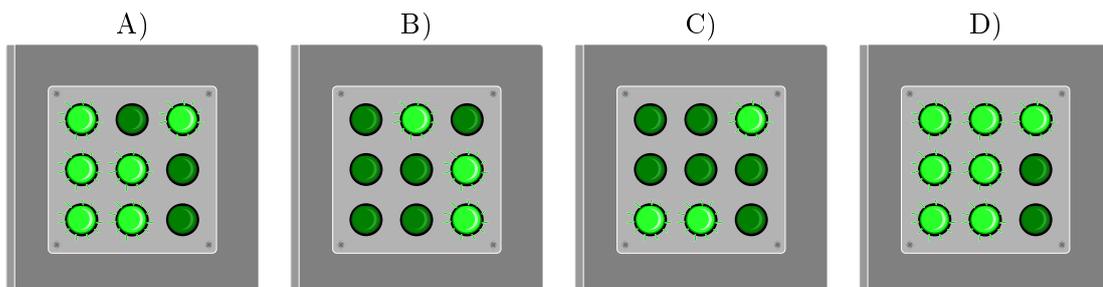
Al momento la combinazione di tasti (sinistra) e i LED accesi (destra) è la seguente:



Erich cambia quindi la combinazione come segue:



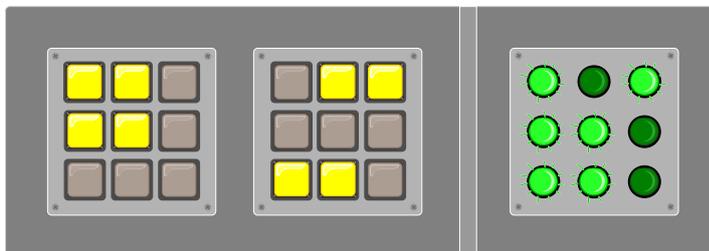
Quali LED si accenderanno sulla destra?





Soluzione

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



Questa è l'informatica!

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

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

Siti web e parole chiave

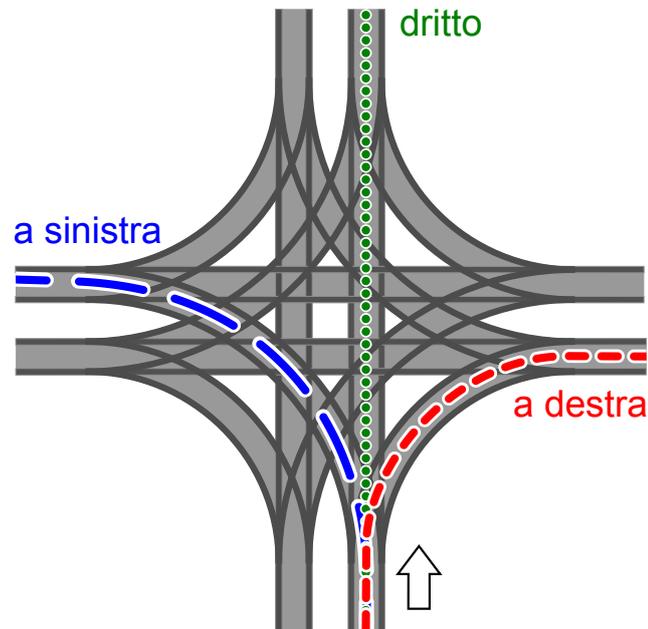
logica binaria, funzione XOR

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



14. Incrocio di tram

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



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

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

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

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



Soluzione

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

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

Questa è l’informatica!

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

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

Siti web e parole chiave

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

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



15. La codifica delle bandiere

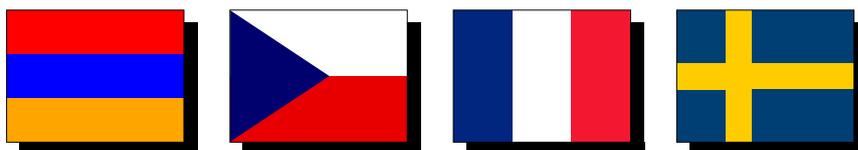
Le immagini del computer sono composte da righe di elementi di immagine (pixel). Nel caso più semplice tali immagini vengono memorizzate come file in cui il colore di ogni pixel è descritto singolarmente. Con il formato per file GIW, le immagini vengono anche compresse e quindi memorizzate in un file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

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

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

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

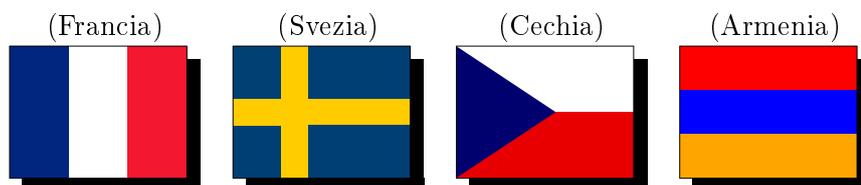
Ordina le immagini per grandezza del file GIW. Trascina le bandiere in modo che a sinistra ci sia la bandiera con il file GIW più grande.





Soluzione

La successione di bandiere esatta è:



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

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

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

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

Questa è l'informatica!

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

Siti web e parole chiave

codifica, compressione, algoritmi di compressione, grafica bitmap

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



A. Autori dei quesiti

 Guðjón Karl Arnarson
 Wilfried Baumann
 Andrea Brabcová
 Eugenio Bravo
 Nicolas Brunner
 Christian Datzko
 Susanne Datzko
 Olivier Ens
 Jürgen Frühwirth
 Haris Gavranovic
 Martin Guggisberg
 Urs Hauser
 Hans-Werner Hein

 Mathias Hiron
 Juraj Hromkovič
 Akiko Kikui
 Tobias Kohn
 Ivana Kosírová
 Greg Lee
 Hiroki Manabe
 Tom Naughton
 Erkulan Nurtazanov
 Henry Ong
 Serena Pedrocchi
 Wolfgang Pohl
 Ilya Posov

 Sergei Pozdniakov
 Dániel Pressing
 Lorenzo Repetto
 Kirsten Schlüter
 Eljakim Schrijvers
 Sue Sentance
 Björn Steffen
 Seiichi Tani
 Jiří Vaníček
 Troy Vasiga
 Michael Weigend



B. Sponsoring: concorso 2016

HASLERSTIFTUNG

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

ROBOROBO

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

digitec.ch

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

**bischof
berger**

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

verkehrshaus.ch

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



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

Standortförderung beim Amt für Wirtschaft und Arbeit
Kanton Zürich



i-factory (Museo Svizzero dei Trasporti, Lucerna)

UBS

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

bbv
Software Services

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

PRESENTEX
Das Geschenk - die gute Werbung

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



ITgirls@hslu

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

PH LUZERN
PÄDAGOGISCHE
HOCHSCHULE

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

ABZ

AUSBILDUNGS- UND BERATUNGSZENTRUM
FÜR INFORMATIKUNTERRICHT

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



C. Ulteriori offerte

010100110101011001001001
 0100000100101110101010011
 010100110100100101000101
 001011010101001101010011
 010010010100100100100001

SSII

www.svia-ssie-ssii.ch
 schweizerischervereinfürinformatikind
 erausbildung//sociétésuisse del'inform
 atique dans l'enseignement//societàsviz
 zeraperl'informaticanell'insegnamento

Diventate membri della SSII <http://svia-ssie-ssii.ch/verein/mitgliedschaft/> sostenendo in questo modo il Castoro Informatico.

Chi insegna presso una scuola dell'obbligo, media superiore, professionale o universitaria in Svizzera può diventare membro ordinario della SSII.

Scuole, associazioni o altre organizzazioni possono essere ammesse come membro collettivo.