



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

Quesiti e soluzioni 2019 5^o e 6^o anno scolastico

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

A cura di:

Lucio Negrini, Christian Datzko, Susanne Datzko, Juraj Hromkovič, Regula Lacher

010**100**1101010**11001001001**
01000**00100**101101010100**11**
0**10100**110100100101000**0101**
00101**101010100**11**101010011**
0**10010010100**100100100**001**

SS! I

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 2019

Christian Datzko, Susanne Datzko, Olivier Ens, Hanspeter Erni, Nora A. Escherle, Martin Guggisberg, Saskia Howald, Lucio Negrini, Gabriel Parriaux, Elsa Pellet, Jean-Philippe Pellet, Beat Trachsler.

Un particolare ringraziamento va a:

Juraj Hromkovič, Michelle Barnett, Michael Barot, Anna Laura John, Dennis Komm, Regula Lacher, Jacqueline Staub, Nicole Trachsler: ETHZ

Gabriel Thullen: Collège des Colombières

Valentina Dagienė: Bebras.org

Wolfgang Pohl, Hannes Endreß, Ulrich Kiesmüller, Kirsten Schlüter, Michael Weigend: Bundesweite Informatikwettbewerbe (BWINF), Germania

Chris Roffey: University of Oxford, Regno Unito

Carlo Bellettini, Violetta Lonati, Mattia Monga, Anna Morpurgo: ALaDDIn, Università degli Studi di Milano, Italia

Gerald Futschek, Wilfried Baumann, Florentina Voboril: Oesterreichische Computer Gesellschaft, Austria

Zsuzsa Pluhár: ELTE Informatikai Kar, Ungheria

Eljakim Schrijvers, Justina Dauksaite, Arne Heijenga, Dave Oostendorp, Andrea Schrijvers, Kyra Willekes, Saskia Zweerts: Cuttle.org, Paesi Bassi

Christoph Frei: Chragokyberneticks (Logo Castoro Informatico Svizzera)

Andrea Leu, Maggie Winter, Brigitte Manz-Brunner: 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 Elsa Pellet mentre quella italiana da Veronica Ostini.



INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Il Castoro Informatico 2019 è stato organizzato dalla Società Svizzera per l'Informatica nell'Insegnamento SSII con il sostegno della fondazione Hasler.

HASLERSTIFTUNG

Nota: Tutti i link sono stati verificati l'01.11.2019. Questo quaderno è stato creato il 2 gennaio 2020 col sistema per la preparazione di testi \LaTeX .



I quesiti sono distribuiti con Licenza Creative Commons Attribuzione – Non commerciale – Condividi allo stesso modo 4.0 Internazionale. Gli autori sono elencati a pagina 40.



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” (<https://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 2019 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

Alla categoria del 3^o e 4^o anno scolastico sono stati assegnati 9 quesiti da risolvere, di cui 3 facili, 3 medi e 3 difficili. Alla categoria del 5^o e 6^o anno scolastico sono stati assegnati 12 quesiti, suddivisi in 4 facili, 4 medi e 4 difficili. Ogni altra categoria ha ricevuto invece 15 quesiti da risolvere, di cui 5 facili, 5 medi e 5 difficili.

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 ottenere buoni risultati scegliendo le risposte in modo casuale.

Ogni partecipante ha iniziato con un punteggio pari a 45 punti (risp., Piccolo Castoro: 27 punti, 5^o e 6^o anno scolastico: 36 punti).

Il punteggio massimo totalizzabile era dunque pari a 180 punti (risp., Piccolo castoro: 108 punti, 5^o e 6^o anno scolastico: 144 punti), 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

Lucio Negrini

<https://www.castoro-informatico.ch/it/kontaktieren/>

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

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



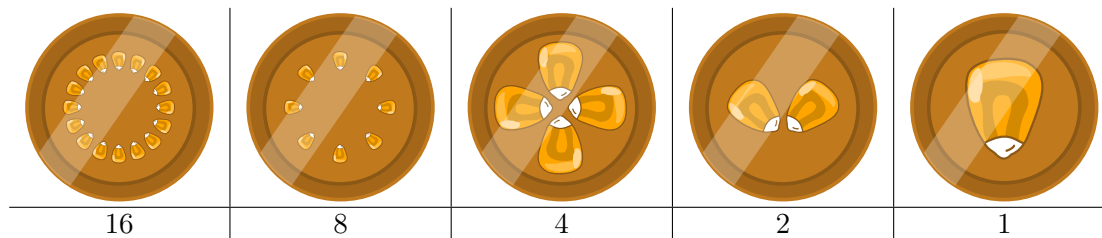
Indice

Hanno collaborato al Castoro Informatico 2019	i
Premessa	ii
Indice	iv
1. Beavercoins	1
2. Segnali di fumo	3
3. Il timbro	5
4. Quale torre?	9
5. Viaggiando nell'universo	11
6. Robot disegnatore	15
7. Rangoli	19
8. Pupazzi di neve e cappelli	21
9. Celebrity-Status	25
10. Messaggio dagli Antichi Castori	29
11. Caratteri cinesi variopinti	33
12. Ingredienti degli hamburger	37
A. Autori dei quesiti	40
B. Sponsoring: concorso 2019	41
C. Ulteriori offerte	43



1. Beavercoins




Nel paese dei castori si usano le “Beavercoins” come valuta. Le monete hanno i seguenti valori:



I castori non si portano dietro volentieri molte monete e quindi pagano con meno monete possibili. Con quali monete pagheresti 13 Beavercoins usando meno monete possibili?



Soluzione

La migliore soluzione, e quindi quella corretta, è di pagare con  ,  e  , quindi con una moneta da 8-Beavercoins, una moneta da 4-Beavercoins e una moneta da 1-Beavercoins. La somma delle monete corrisponde a $8 + 4 + 1 = 13$. Con meno monete non è possibile, perché una moneta più grossa della moneta da 8-Beavercoins sarebbe già quella da 16-Beavercoins e non c'è nessuna moneta del valore dei mancanti 5 Beavercoins. La seconda moneta più piccola è quella da 4-Beavercoins, quindi si ha bisogno, assieme alla moneta da 1-Beavercoins, proprio di queste tre monete.

Per trovare la soluzione corretta si può anche iniziare con un'altra combinazione, ad esempio con due monete da 4-Beavercoins, una da 2-Beavercoins e tre da 1-Beavercoins. Successivamente si possono sostituire due monete dello stesso valore con una moneta di valore doppio fino a quando si raggiunge il risultato corretto.

Questa è l'informatica!

Informatiche e informatici sono esperti nel rappresentare informazioni come sequenze di simboli. Ciò include anche la rappresentazione dei numeri. In questo problema si tratta di una somma di denaro che può essere pagata con diverse combinazioni di monete. Questa combinazione non è unica, diverse combinazioni con monete di diverso valore danno la stessa somma di denaro. Quindi in questo problema si tratta anche di trovare la combinazione con il minor numero di monete.

In questo problema le monete sono scelte in modo che il valore di due monete con lo stesso valore assieme corrisponde sempre a quello della prossima moneta più grossa. Il risultato è il *sistema binario* con i valori 1, 2, 4, 8, 16 eccetera. Nel sistema numerico binario la rappresentazione di qualsiasi numero come 13 è sempre unica: un valore è utilizzato oppure no.

Anche l'abaco, una macchina per calcolare che si usava molte centinaia di anni fa e che viene usata ancora oggi nell'epoca delle calcolatrici in alcune regioni del mondo, funziona in modo simile.

Parole chiave e siti web





Sistema numerico binari, Abaco

- https://it.wikipedia.org/wiki/Sistema_numerico_binario
- <https://it.wikipedia.org/wiki/Abaco>



2. Segnali di fumo

Un castoro si siede sempre sulla montagna e osserva il tempo. Dice ai castori nella valle come sarà il tempo. Usa segnali di fumo che consistono in cinque nuvole di fumo successive. Una nuvola di fumo o è piccola o è grande. I castori hanno concordato i seguenti segnali di fumo:

			
Sarà temporalesco	Sarà piovoso	Sarà nuvoloso	Sarà soleggiato

In un giorno ventoso i castori nella valle non riescono a riconoscere bene le nuvole di fumo. Sono solo sicuri che la seconda e la quarta nuvola di fumo sono grandi, le altre le hanno sostituite con dei punti di domanda:



Cosa potrebbe significare?

- A) “Sarà temporalesco” o “Sarà piovoso”.
- B) “Sarà piovoso” o “Sarà nuvoloso”.
- C) “Sarà piovoso” o “Sarà soleggiato”.
- D) “Sarà temporalesco” o “Sarà nuvoloso”.



Soluzione





I castori nella valle hanno riconosciuto alla seconda e alla quarta posizione una nuvola di fumo grande. Anche per i segnali di fumo “Sarà temporalesco” e “Sarà nuvoloso” ci sono alla seconda e alla quarta posizione delle nuvole grandi. Per “Sarà piovoso” e “Sarà soleggiato” in queste posizioni ci sono delle nuvole di fumo piccole, quindi questi segnali di fumo non coincidono con l’osservazione dei castori nella valle.

Quindi la risposta corretta è la D) “Sarà temporalesco” o “Sarà nuvoloso”.

Questa è l’informatica!

Quando si vuole trasmettere un messaggio si vuole che arrivi correttamente al destinatario. In questo problema il messaggio viene trasmesso con l’aiuto di nuvole di fumo grandi e piccole. Nel caso generale si parla di *simboli*. È quindi sensato scegliere una sequenza di simboli in modo che il messaggio trasmesso sia comprensibile anche se viene danneggiato durante la trasmissione. Ciò si può ottenere comunicando più informazioni di quelle strettamente necessarie. Queste informazioni aggiuntive vengono chiamate *ridondanti*.

Quando si riesce a ricostruire l’informazione danneggiata con al massimo n errori, si parla di codificazione n -autoregolante. Rappresentare messaggi come sequenze di simboli in modo da potere ricostruire il messaggio, anche quando la sua rappresentazione viene danneggiata durante la trasmissione, è un compito tipico degli informatici. Ad esempio rendono possibile riprodurre correttamente musica da dei CD o video da dei DVD anche quando nella trasmissione si presentano alcuni errori. D’altronde, per questo problema sarebbero abbastanza due nuvole di fumo per trasmettere i quattro diversi messaggi:

			
Sarà temporalesco.	Sarà piovoso.	Sarà nuvoloso.	Sarà soleggiato.

I castori, però, usano cinque nuvole di fumo. Ciò permette loro di capire i messaggi correttamente anche in casi dove due nuvole di fumo, o addirittura in certi casi tre nuvole di fumo, sono “illeggibili”. D’altronde, i castori hanno pensato ai messaggi in modo tale che ogni due messaggi si differenziano in almeno tre posizioni.

Parole chiave e siti web

Rilevazione e correzione d’errore

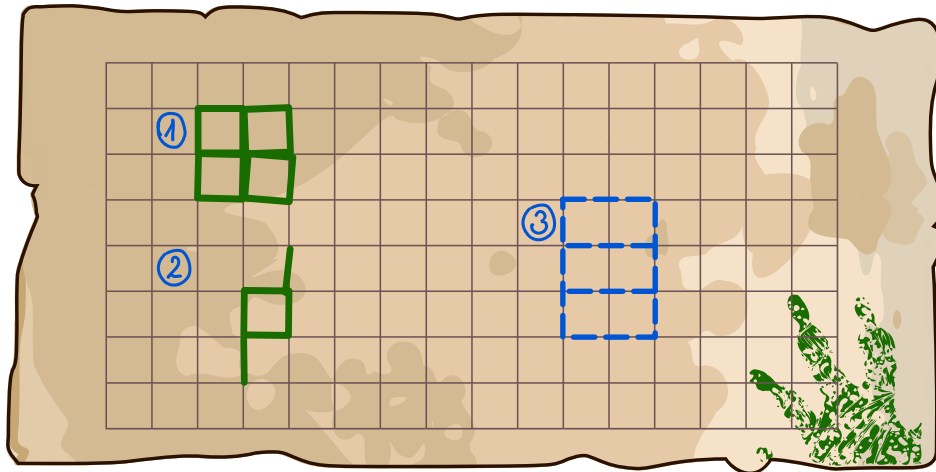
- https://it.wikipedia.org/wiki/Rilevazione_e_correzione_d'errore



3. Il timbro

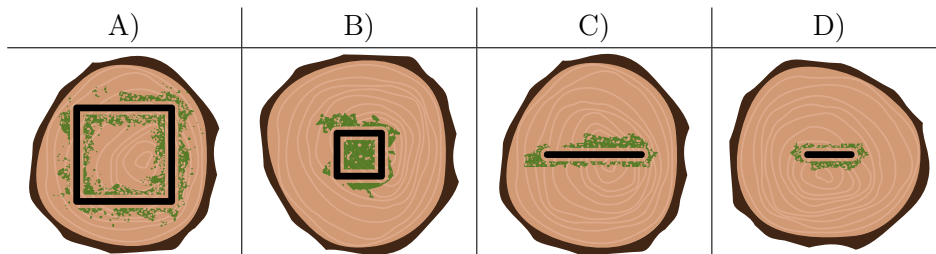
Il castoro Paul ha quattro timbri A, B, C e D, come mostrato sotto. Con questi timbri Paul ha timbrato le due figure ① e ②.

- Per la figura ① Paul ha usato quattro volte il timbro B.
- Per la figura ② Paul ha usato una volta il timbro B e due volte il timbro D.



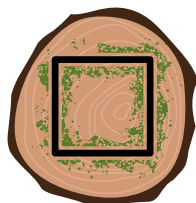
Adesso Paul vuole avere la figura ③. Maria, la sorella di Paul, afferma che lei per creare la figura deve solo timbrare due volte.

Che timbro utilizzerebbe Maria?

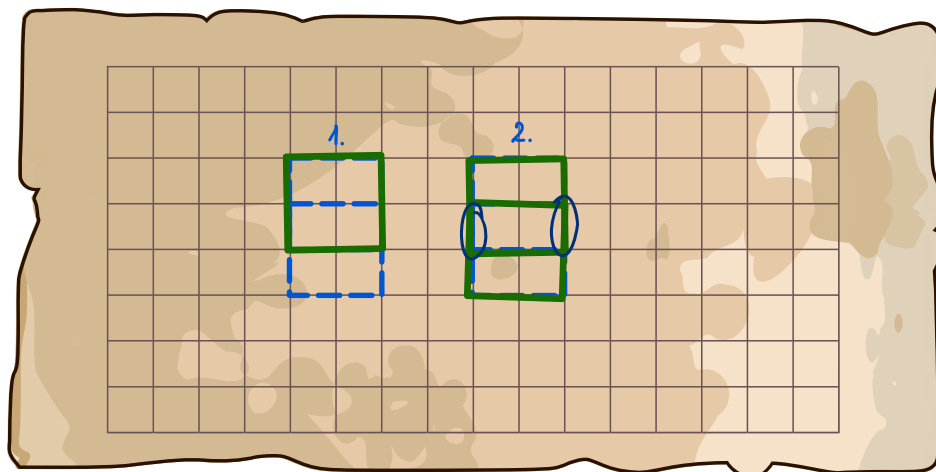




Soluzione



La risposta corretta è A) . Se Maria ha timbrato un quadrato grande e sposta il timbro di un quadretto verso l'alto o verso il basso si forma esattamente la figura desiderata. Due parti delle linee si sovrappongono, ma se lei timbra in modo preciso non si vede:



Con gli altri timbri non si può ottenere la figura desiderata:

- Con il timbro B è impossibile timbrare un rettangolo largo due quadretti senza linea centrale.
- Con il timbro C potrebbe sicuramente timbrare la figura, ma poiché la figura ha linee della lunghezza totale di quattordici quadretti e timbrando una volta sola si copre solo la lunghezza di due quadretti, ha bisogno di timbrare almeno sette volte. Se si guarda attentamente si può constatare che ha perfino bisogno di timbrare otto volte, perché per riuscire a timbrare entrambi i tratti verticali ha bisogno di timbrare due volte (con sovrapposizioni) in aggiunta alle tre linee orizzontali.
- Con il timbro D potrebbe sicuramente timbrare la figura, ma poiché la figura ha linee della lunghezza totale di quattordici quadretti e timbrando una volta sola può timbrare solo la lunghezza di un quadretto, ha quindi bisogno di timbrare quattordici volte.

Questa è l'informatica!

Per molti problemi ci sono molte soluzioni diverse che portano tutte all'obiettivo. Spesso alcune sono veloci da trovare, come per esempio una soluzione con l'aiuto del timbro C o D. Ma non tutte le soluzioni sono "buone" allo stesso modo: le diverse soluzioni si differenziano, ad esempio, nel numero di volte che bisogna timbrare.

Uno dei compiti dell'informatica è trovare tra molte soluzioni di un problema quella "migliore". Questo è molto importante nella pratica: quando si può risolvere un problema in un'ora al posto che in un giorno, restano molte ore per potersi occupare di altri problemi.

Per misurare l'efficienza gli informatici analizzano i processi e descrivono la loro durata in base alla quantità o alla grandezza dei dati da elaborare. Quando ad esempio si cerca una voce in un array



ordinato con 1'000'000 di voci o si può guardare array per array e fare in media 500'000 confronti, oppure si può iniziare in mezzo e continuare a cercare sempre nella rispettiva metà ...e dopo al massimo 20 confronti si è trovato l'elemento! Per 3 secondi a confronto questa sarebbe una differenza tra circa 17 giorni di ricerca ininterrotta o 1 minuto.

Parole chiave e siti web

Efficienza, Teoria della complessità computazionale

- [https://it.wikipedia.org/wiki/Efficienza_\(informatica\)](https://it.wikipedia.org/wiki/Efficienza_(informatica))
- https://it.wikipedia.org/wiki/Teoria_della_complessit%C3%A0_computazionale

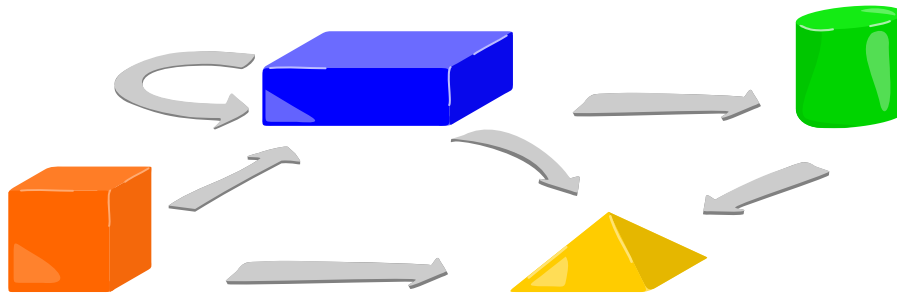




4. Quale torre?

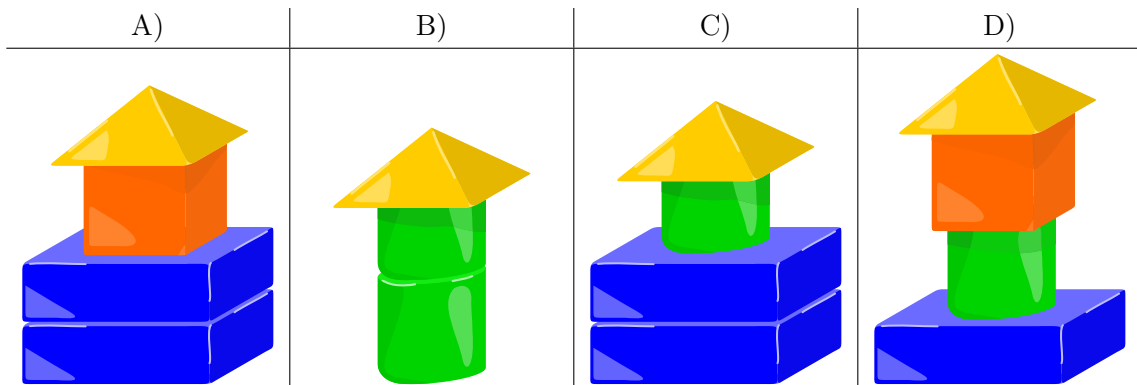
La sorellina piccola di Leon ha formulato delle regole su come i blocchetti del gioco delle costruzioni possono essere impilati. Queste regole sono state illustrate con delle frecce in un disegno. Inoltre vale:

- Leon può cominciare con qualsiasi blocchetto.
- Leon deve sempre seguire le frecce. Se da un blocchetto partono molte frecce lui ne può scegliere una. Quando una freccia indica indietro verso lo stesso blocchetto può impilarne un altro di quel tipo sulla torre.
- Leon deve smettere quando da un blocchetto appena impilato non parte più nessuna freccia.



Leon impila quattro torri diverse per la sua sorellina piccola.

Quale delle quattro torri ha costruito secondo le regole della sua sorellina piccola?





Soluzione

La torre della risposta A) inizia correttamente con due parallelepipedi blu. Dopo il secondo parallelepipedo blu segue tuttavia un cubo arancione, ma non c'è nessuna freccia dal parallelepipedo blu verso il cubo arancione. Quindi la risposta A) è sbagliata.

La torre della risposta B) inizia correttamente con un cilindro verde. Dopo il cilindro verde segue tuttavia un ulteriore cilindro verde, ma non c'è nessuna freccia dal cilindro verde indietro verso il cilindro verde. Quindi la risposta B) è sbagliata.

La torre della risposta C) inizia correttamente con due parallelepipedi blu. Dopo il secondo parallelepipedo blu segue correttamente un cilindro verde e dopo il cilindro verde segue correttamente una piramide gialla. Siccome non parte nessuna freccia dalla piramide gialla è corretto che non venga impilato sopra nessun altro blocchetto. Quindi la risposta C) è giusta.

La torre della risposta D) inizia correttamente con un parallelepipedo blu. Dopo il parallelepipedo blu segue correttamente un cilindro verde. Dopo il cilindro verde segue tuttavia un cubo arancione, ma non c'è nessuna freccia dal cilindro verde al cubo arancione. Quindi la risposta D) è sbagliata.

Questa è l'informatica!

Le regole per la costruzione di una torre si basano sul fatto che il blocchetto più in alto della torre decide sempre quali blocchetti sono permessi come prossimi blocchetti. Quindi il blocchetto più in alto della torre è lo *stato attuale* della torre. Le regole stabiliscono in quale prossimo stato può *trasformarsi* la torre. Il grafico con le frecce è quindi un *diagramma di stato* o anche *diagramma di transizione di stato*. Siccome tutti i blocchetti possono essere usati come blocchetti più in basso, tutti i blocchetti sono possibili *stati iniziali*. La piramide gialla è l'unico blocchetto che è uno *stato finale*, con il quale la torre viene terminata (se prima non è caduta). La decisione di impilare un altro blocchetto sulla torre è un *input* del costruttore.

Questi aspetti della costruzione di una torre descrivono un cosiddetto *automa a stati finiti non deterministico*. Si chiama non deterministico perché ci sono stati nei quali possono essere scelte diverse prossime strade: dopo un parallelepipedo blu possono seguire un ulteriore parallelepipedo blu, un cilindro verde o una piramide gialla. Si chiama finito perché c'è solo un insieme finito di possibili stati: uno dei quattro blocchetti come blocchetto superiore della torre. Teoreticamente, però, con esso si potrebbe costruire una torre infinita ... ma per farlo ci sarebbe bisogno di infiniti parallelepipedi blu e le torri alte a volte cadono (spesso per la grande gioia dei costruttori).

Il modello dell'automa a stati finiti non deterministico viene usato spesso in informatica. Si adatta bene alla descrizione di cose molto diverse: il comportamento di moduli di software o anche interi programmi, facili strutture linguistiche, l'interazione di componenti di un hardware e molto altro ancora. Con l'aiuto di una tale descrizione formale può anche essere testato se il software funziona come desiderato ... o se la torre è costruita in modo giusto.

Parole chiave e siti web

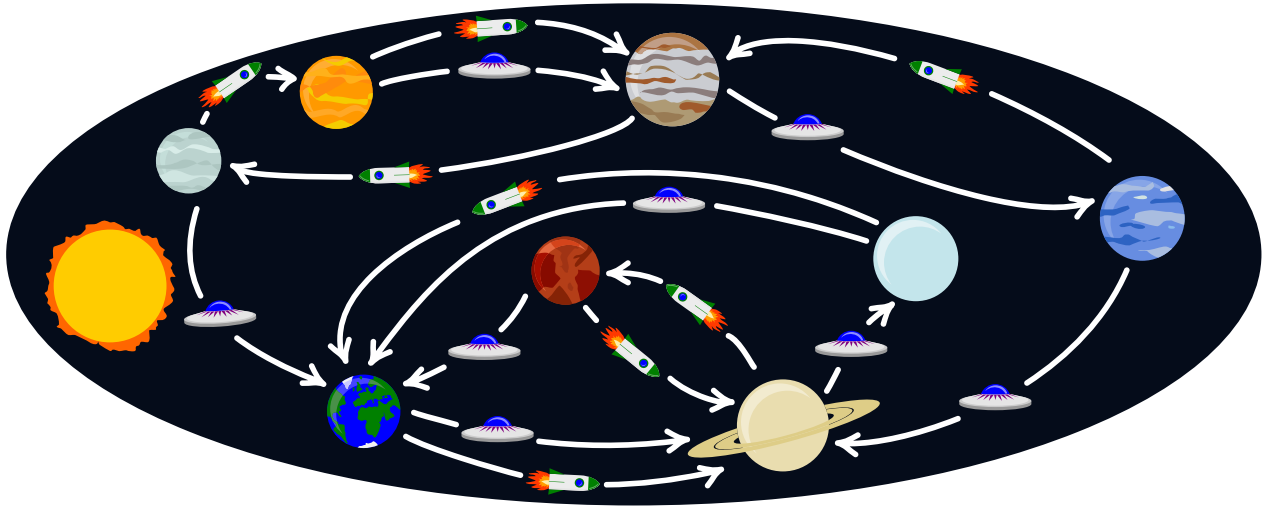
Automa a stati finiti non deterministico

- https://it.wikipedia.org/wiki/Automa_a_stati_finiti_non_deterministico
- <https://www.swisseduc.ch/informatik/karatojava/kara/index.html>
- <https://educ.ethz.ch/unterrichtsmaterialien/informatik/kara.html>



5. Viaggiando nell'universo

Gli astronauti possono volare tra i pianeti del nostro sistema solare con dei razzi 🚀 o degli UFO 🛸. La carta seguente raffigura le possibili rotte aeree:



Un astronauta che vuole viaggiare da Venere 🟠 a Saturno 🪐 può volare con un razzo 🚀 o con un UFO 🛸 fino a Giove 🪛. Poi può volare con un UFO 🛸 fino a Nettuno 🌌 e alla fine con un UFO 🛸 fino al suo pianeta di destinazione Saturno 🪐. Se l'astronauta vola prima con un razzo e poi con due UFO, si descrive il viaggio in questo modo:
















L'astronauta Heidi al momento è sul pianeta Nettuno 🌌 e vuole tornare sulla Terra 🌍. L'agenzia viaggi dello spazio le invia quattro proposte.

Quale delle quattro proposte non porta Heidi sulla Terra 🌍?








- A) 🛸 🛸 🚀
- B) 🚀 🛸 🚀 🛸
- C) 🚀 🛸 🛸 🛸 🚀
- D) 🚀 🚀 🚀 🛸














Soluzione

La risposta B)     è l'unica sbagliata. Se Heidi segue questa proposta atterra alla fine ancora su Nettuno . Infatti, prima vola con un razzo  fino a Giove , poi con un UFO  indietro fino a Nettuno , poi ancora con un razzo  fino a Giove  alla fine ancora con un UFO  indietro fino a Nettuno .



Le altre tre proposte portano tutte indietro fino alla Terra . Le stazioni sono:

Risposta A): Da Nettuno  con un UFO  fino a Saturno , con un UFO  fino a Urano  e con un razzo  fino alla Terra .

Risposta C) Da Nettuno  con un razzo  fino a Giove , con un UFO  fino a Nettuno , con un UFO  fino a Saturno , con un UFO  fino a Urano  e con un razzo  fino alla Terra .

Risposta D): Da Nettuno  con un razzo  fino a Giove , con un razzo  fino a Mercurio  e con un UFO  fino alla Terra .

Questa è l'informatica!

La carta delle possibili rotte aeree da pianeta a pianeta ha una proprietà particolare: da ogni pianeta partono esattamente due rotte aeree, una con un razzo  e una con un UFO . Quindi è sempre chiaro su quale pianeta si vola quando viene indicato con quale mezzo di trasporto si vola.

Una carta di questo tipo descrive un *automa a stati finiti deterministico*. Un automa di questo tipo consiste in un insieme di possibili *stati* (in questo caso sono i nomi dei pianeti come posizione dell'astronauta), un insieme di possibili *transizioni di stato* (in questo caso sono le frecce sulla carta con il cui aiuto l'astronauta può cambiare la sua posizione) basati su determinati *input* ("razzo" o "UFO") come anche uno *stato iniziale* (in questo caso "Nettuno") e un insieme di *stati finiti* (in questo caso solo lo stato "Terra"). La carta si chiama anche un *diagramma di stato* o un *diagramma di transizione di stato*.

Gli automi a stati finiti deterministici vengono usati in diversi ambiti perché sono molto facili da programmare. Esempi tipici sono macchine del caffè, lavastoviglie o anche distributori di bevande. Ma vengono anche utilizzati per riconoscere delle parole correttamente (ad esempio se un testo inserito rappresenta un indirizzo e-mail). Gli automi a stati finiti deterministici si possono anche associare con una determinata classe di grammatiche (cosiddette *grammatiche regolari*) e una determinata classe di linguaggi artificiali (cosiddetti *linguaggi regolari*) e saltare da un "mondo" all'altro. Questo aiuta a risolvere molti problemi.

D'altronde l'agenzia viaggi dello spazio ha un altro compito: deve trovare nel diagramma di stato un possibile percorso da uno stato all'altro. Per questo aiuta vedere il diagramma di stato come *grafo orientato* e cercare un possibile percorso da un *nodo* a un altro nodo con aiuto degli *archi* dati. Per questo ci sono degli algoritmi standard, in modo che l'agenzia viaggi dello spazio non deve iniziare a cercare sempre da capo.

Parole chiave e siti web

Automa a stati finiti deterministico, Grafi

- https://it.wikipedia.org/wiki/Automa_a_stati_finiti_deterministico
- <https://www.swisseduc.ch/informatik/karatojava/kara/index.html>
- <https://educ.ethz.ch/unterrichtsmaterialien/informatik/kara.html>



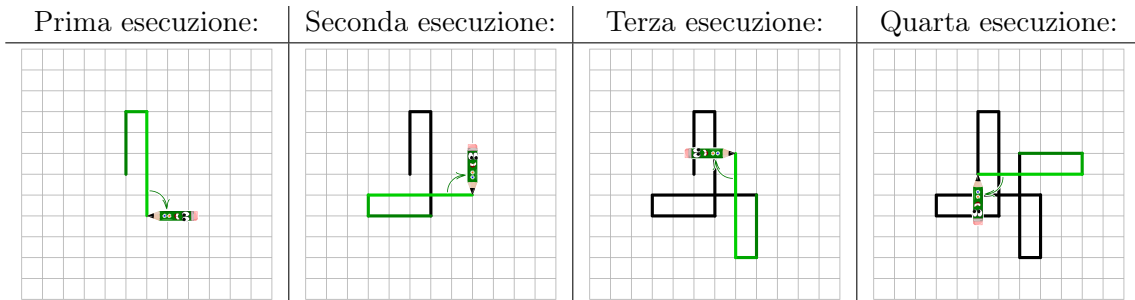
- <https://it.wikipedia.org/wiki/Grafo>





6. Robot disegnatore

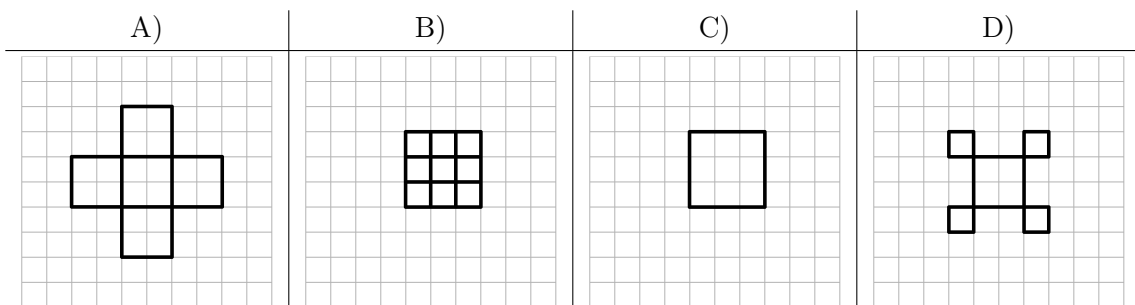
Un robot si muove su una griglia e disegna delle linee. Può essere pilotato con l'aiuto di tre numeri. Se gli si dà i numeri $3\curvearrowright 1\curvearrowright 5\curvearrowright$ disegna la figura seguente:



In questo caso ripete i seguenti passaggi quattro volte:

- Va in avanti sulla griglia di quanti quadretti sono dati dal primo numero.
- Si ruota di un quarto di giro verso destra.
- Va in avanti sulla griglia di quanti quadretti sono dati dal secondo numero.
- Si ruota di un quarto di giro verso destra.
- Va in avanti sulla griglia di quanti quadretti sono dati dal terzo numero.
- Si ruota di un quarto di giro verso destra.

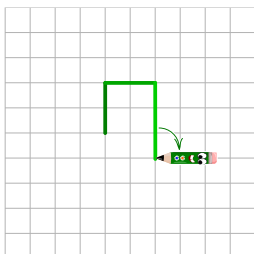
Al robot vengono dati i numeri $2\curvearrowright 2\curvearrowright 3\curvearrowright$. Come appaiono le linee disegnate?



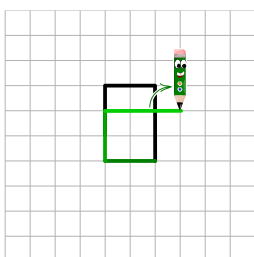


Soluzione

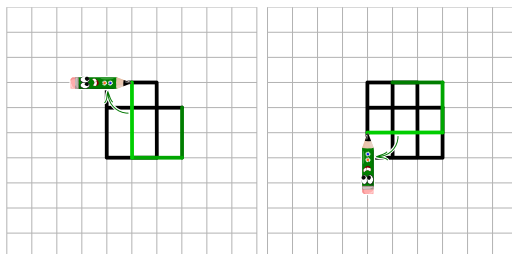
I numeri $2\curvearrowright 2\curvearrowright 3\curvearrowright$ significano che prima il robot va in avanti di 2 quadretti, si ruota di un quarto di giro verso destra, va ancora avanti di 2 quadretti, si ruota di un quarto di giro verso destra, poi va avanti di 3 quadretti, e si ruota ancora di un quarto di giro verso destra. Quindi ha disegnato le linee seguenti:



Quando ciò viene ripetuto ha disegnato in totale le linee seguenti:

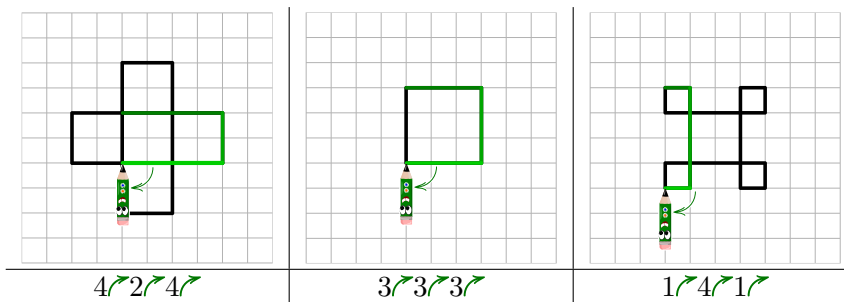


Dopo due ripetizioni l'immagine appare così:



Quindi la risposta B) è corretta.

D'altronde le altre tre figure si lasciano anche disegnare con l'aiuto del robot, necessitano semplicemente di altri numeri:





Questa è l'informatica!

Il robot disegnatore in questo problema può eseguire soltanto programmi molto semplici. Il linguaggio di programmazione che capisce il robot, riconosce solo numeri come istruzioni. Ogni programma deve consistere in esattamente questi tre numeri seguiti dal simbolo ↗. Inoltre è incorporato in modo fisso che il robot ripete le istruzioni contenute nel programma quattro volte, sia che se lo si voglia o meno.

La maggior parte dei robot e dei computer comprende linguaggi (di programmazione) molto più complessi. La maggior parte di questi linguaggi ha le stesse caratteristiche di base:

1. I programmi possono essere costituiti da un numero qualsiasi di istruzioni che vengono eseguite una dopo l'altra come una *sequenza*.
2. Istruzioni ripetute di diverso tipo, i cosiddetti *cicli*, possono, ma non devono, essere utilizzati.
3. Inoltre, ci sono istruzioni condizionali che consentono diverse esecuzioni del programma a seconda dello stato del programma, le cosiddette *diramazioni*.

Se un linguaggio di programmazione contiene cicli e diramazioni, con il suo aiuto si può calcolare tutto ciò che è calcolabile. In informatica tali linguaggi di programmazione vengono designati come universali – o anche come *Turing equivalenti*.

D'altronde il robot in questo problema è un classico contesto nel quale si può imparare a programmare. Al posto del robot ci si può immaginare una tartaruga che disegna delle linee. Tali *grafiche della tartaruga* possono essere compilate ad esempio con XlogoOnline o anche con TigerJython.

Parole chiave e siti web

Grafica della tartaruga

- https://it.wikipedia.org/wiki/Programmazione_strutturata
- https://it.wikipedia.org/wiki/Turing_equivalenza
- https://en.wikipedia.org/wiki/Turtle_graphics
- <https://xlogo.inf.ethz.ch/>
- <http://www.tigerjython.ch/>

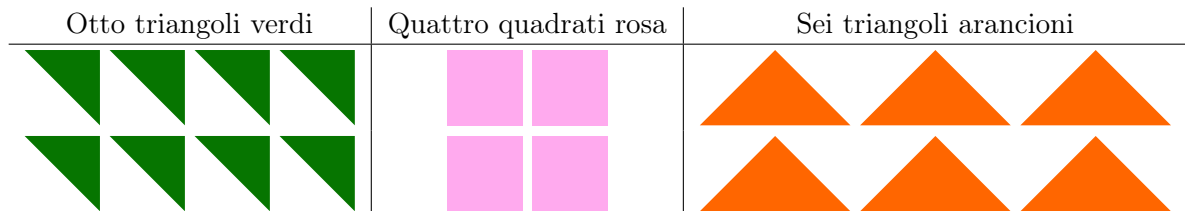




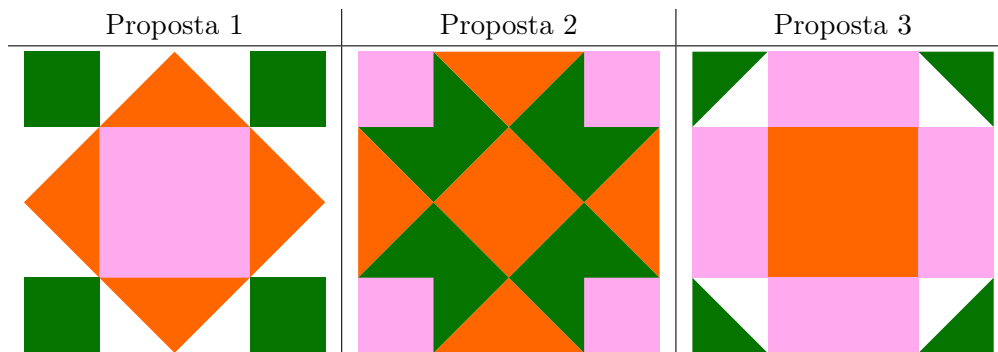
7. Rangoli

Il rangoli è una forma d'arte proveniente dall'India. Vengono posati dei motivi sul pavimento. Questi motivi sono per lo più simmetrici.

Per il suo rangoli Priya ha delle pietre di tre forme diverse: otto triangoli verdi, quattro quadrati rosa e sei triangoli arancioni. Le pietre dello stesso colore hanno anche la stessa grandezza:



Priya trova le seguenti proposte per dei rangoli su un sito internet (le superfici bianche rimangono libere):



Quali delle tre proposte per dei rangoli può posare Priya con le sue pietre?

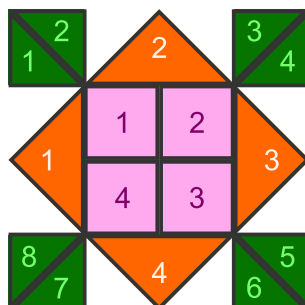
- A) Solo la proposta 1.
- B) Solo la proposta 2.
- C) Solo la proposta 3.
- D) Tutte e tre le proposte.





Soluzione

Priya può posare A) solo la proposta 1 con le sue pietre.

La grafica seguente conta le pietre di diverso tipo nella proposta 1. Poiché ha bisogno al massimo di tante pietre di ogni tipo quante ne ha a sua disposizione, può posare la proposta 1:



Per la proposta 2 avrebbe bisogno in totale di dodici triangoli verdi, poiché ognuna delle quattro figure verdi nella proposta 2  necessita tre triangoli verdi. Priya ha però solo otto triangoli verdi a disposizione, quindi non può posare la proposta 2.

Per la proposta 3 avrebbe bisogno in totale di otto quadrati rosa, poiché ognuna delle quattro figure rosa nella proposta 3  necessita 2 quadrati rosa. Priya ha però solo quattro quadrati rosa a disposizione, quindi non può posare la proposta 3.

Siccome non può posare né la proposta 2 né la proposta 3, neanche la risposta D) può essere corretta

Questa è l'informatica!

Il *rangoli* è una forma d'arte che è tradizionalmente realizzata in India con riso e farina colorati, ma anche con sabbia colorata o fiori. I rangoli hanno principalmente uno scopo decorativo, ma sono anche legati a tradizioni regionali o di famiglia e agli auguri di buon auspicio. Anche alcune tradizioni religiose sono legate ai rangoli.

In questo problema bisognava scomporre una forma complessa in forme più piccole, che si potevano poi confrontare con le forme di base disponibili. Poi si sa quante di tutte le forme di base si hanno bisogno. Questo processo si chiama *decomposizione*, e viene usato spesso in informatica.

Confrontare le forme scomposte con le forme di base si chiama *Pattern Matching* (ingl. per *correlazione di motivi* o *confronto di motivi*). Nell'informatica il Pattern Matching è di grande importanza, dove non si cercano solo motivi grafici ma anche per esempio parole in testi o nomi di file nei sistemi di file, o anche per il confronto delle sequenze del genoma nella ricerca dei criminali.

Parole chiave e siti web

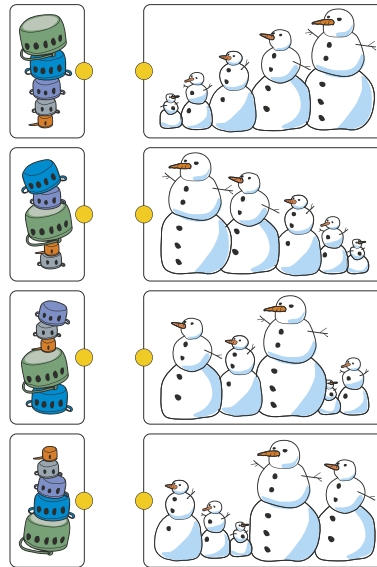
Decomposizione, Pattern Matching

- <https://en.wikipedia.org/wiki/Rangoli>
- https://it.wikipedia.org/wiki/Pattern_matching
- [https://en.wikipedia.org/wiki/Decomposition_\(computer_science\)](https://en.wikipedia.org/wiki/Decomposition_(computer_science))

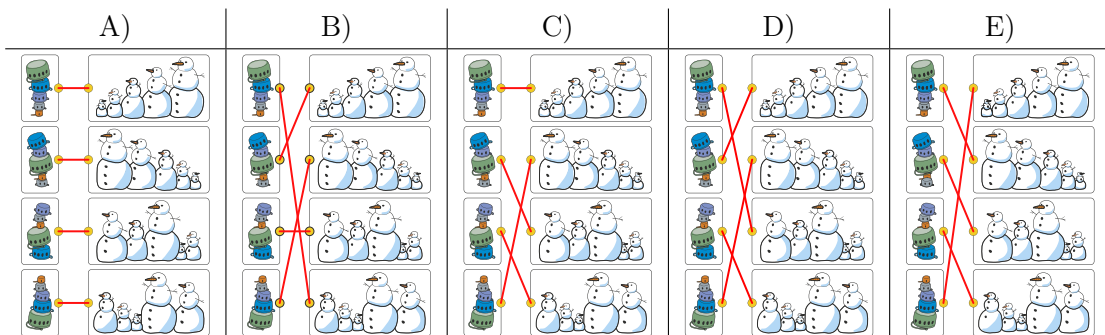


8. Pupazzi di neve e cappelli

Da sinistra a destra vengono distribuiti a cinque pupazzi di neve cinque cappelli prendendoli dalla pila dall'alto al basso. Alla fine, ogni pupazzo di neve dovrebbe ricevere un cappello della sua grandezza.

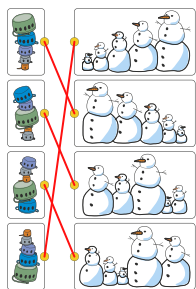


Quale pila di cappelli appartiene a quale fila di pupazzi di neve?





Soluzione



La risposta corretta è la E)

La prima pila di cappelli appartiene alla seconda fila di pupazzi di neve. Il primo pupazzo di neve è il pupazzo di neve più grande e il primo cappello è il cappello più grande. Il secondo pupazzo di neve è il secondo più grande e riceve il secondo cappello più grande e così via.

La seconda pila di cappelli appartiene alla terza fila di pupazzi di neve. Il primo pupazzo di neve è il secondo pupazzo di neve più grande e il primo cappello è il secondo cappello più grande. Anche i rimanenti pupazzi di neve ricevono ognuno il loro cappello corrispondente.

La terza pila di cappelli appartiene alla quarta fila di pupazzi di neve. Il primo pupazzo di neve ottiene il terzo cappello più grande, che è il più in alto. Anche qua i rimanenti pupazzi di neve ricevono ognuno il loro cappello corrispondente.

La quarta pila di cappelli appartiene alla prima fila di pupazzi di neve. Qui i pupazzi di neve sono ordinati da sinistra a destra dal più piccolo al più grande. Anche i cappelli sono ordinati dall'alto al basso dal più piccolo al più grande. Quindi anche in questo caso ogni pupazzo di neve ottiene il suo cappello corrispondente.

La risposta A) non è corretta, siccome nella prima fila il pupazzo di neve più piccolo riceverebbe il cappello più grande.

La risposta B) non è corretta, siccome il pupazzo di neve più piccolo della prima fila riceverebbe il secondo cappello più grande.

La risposta C) non è corretta, siccome come nella risposta A) il pupazzo di neve più piccolo della prima fila riceverebbe il cappello più grande.

La risposta D) non è corretta, siccome come nella risposta B) il pupazzo di neve più piccolo riceverebbe il secondo cappello più grande.

Questa è l'informatica!

Quando come in questo problema si assegna ad ogni pupazzo il "suo" cappello senza modificare l'ordine di grandezza dei pupazzi di neve o dei cappelli, questa viene chiamata una *funzione che mantiene la struttura*. Per ogni fila di pupazzi di neve e per ogni pila di cappelli sono definiti un primo, secondo, terzo, quarto e quinto elemento, i quali devono adattarsi a un collegamento.

Si può vedere l'ordine di grandezza dei cappelli e dei pupazzi di neve come *ennupla*: un insieme ordinato di elementi. In questo caso si considera soltanto la caratteristica interessante, cioè la grandezza dei pupazzi di neve e rispettivamente dei cappelli. Ad esempio le grandezze dei cappelli della prima pila sono (5, 4, 3, 2, 1) e le grandezze dei pupazzi di neve della prima fila sono quindi (1, 2, 3, 4, 5). Se un collegamento con la prima fila funzionasse, le ennuple dovrebbero essere uguali. I cappelli sono rappresentati come una *pila* (ingl. *stack*). Una pila è una struttura di dati con la quale si può fare molto poco (le cosiddette *operazioni*): *rimuovere l'elemento superiore* (ingl. *pop*) e *inserire un elemento sopra* (ingl. *push*). A volte nelle pile si può anche *guardare l'elemento superiore* (ingl. *top*) e *guardare, se la pila è vuota* (ingl. *empty*).

I pupazzi di neve sono rappresentati come una *coda* (ingl. *queue*). Una coda è anche una struttura di dati con la quale si può fare molto poco: *rimuovere l'elemento più in avanti* (ingl. *dequeue*) e



inserire un elemento dietro (ingl. *enqueue*). A volte nelle code si può anche *guardare l'elemento più in avanti* (ingl. *first*) e guardare, *se la coda è vuota* (ingl. *empty*).

Quindi si può rimuovere ogni elemento e confrontare: non appena cappello e pupazzo di neve non coincidono, la soluzione non può più essere corretta. Questo argomento viene usato nella spiegazione della soluzione: per ognuna delle risposte sbagliate viene spiegato perché il primo pupazzo di neve della prima fila di pupazzi di neve non corrisponde al primo cappello della pila che gli è stata assegnata.

D'altronde non sappiamo esattamente se una pila di cappelli è veramente una pila: potrebbe anche essere che ognuno dei cappelli è stato messo sotto la pila (in modo da essere in realtà una coda). Allo stesso modo i pupazzi di neve potrebbero esser "spinti davanti" quindi messi dal davanti (in modo che la fila di pupazzi di neve funzioni come una pila).

Parole chiave e siti web

Funzione che mantiene la struttura, pila, coda

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



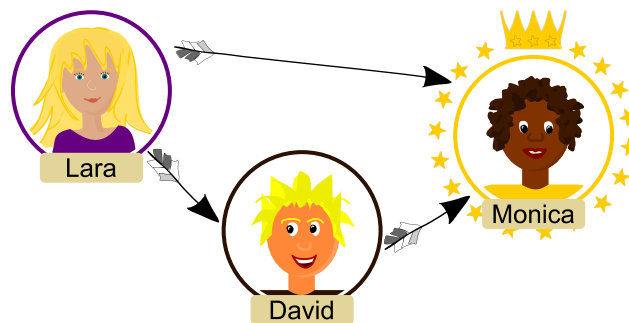


9. Celebrity-Status

Nel social network TeeniGram i membri possono seguire altri membri. Inoltre, in TeeniGram ci sono gruppi di membri. In un gruppo, un membro è una celebrità se ...

- ...la celebrità è seguita da tutti gli altri membri del gruppo e ...
- ...lei stessa non segue nessuno del gruppo.

Nel gruppo seguente Lara segue Monica e David, David segue Monica, ma Monica non segue nessuno. Quindi Monica è una celebrità:



Un altro gruppo è costituito da sei membri: Andrea, Dican, Françoise, Gianni, Robin e Stefan. Si seguono l'uno con l'altro in questo modo:

- Andrea segue Dican, Françoise e Gianni.
- Dican segue Françoise, Gianni e Robin.
- Françoise segue Gianni.
- Robin segue Dican, Françoise e Gianni.
- Stefan segue Andrea, Dican, Françoise, Gianni e Robin.

C'è una celebrità in questo gruppo?

- Sì, Françoise è una celebrità in questo gruppo.
- Sì, Gianni è una celebrità in questo gruppo.
- Sì, Stefan è una celebrità in questo gruppo.
- Sì, Françoise e Gianni sono entrambi celebrità in questo gruppo.
- No, questo gruppo non ha celebrità.



Soluzione

La risposta corretta è la B) “Sì, Gianni è una celebrità in questo gruppo”.

Entrambe le condizioni sono soddisfatte:

- Tutti gli altri membri del gruppo seguono Gianni.
- Gianni stesso non segue nessuno del gruppo.

Le altre risposte sono tutte sbagliate.

La risposta A) non può essere corretta, siccome Françoise stesso segue Gianni. Inoltre Gianni non segue Françoise.

La risposta C) non può essere corretta. Stefan è perfino qualcosa come un’anti-celebrità del gruppo: segue tutti gli altri membri del gruppo, ma nessuno del gruppo lo segue.

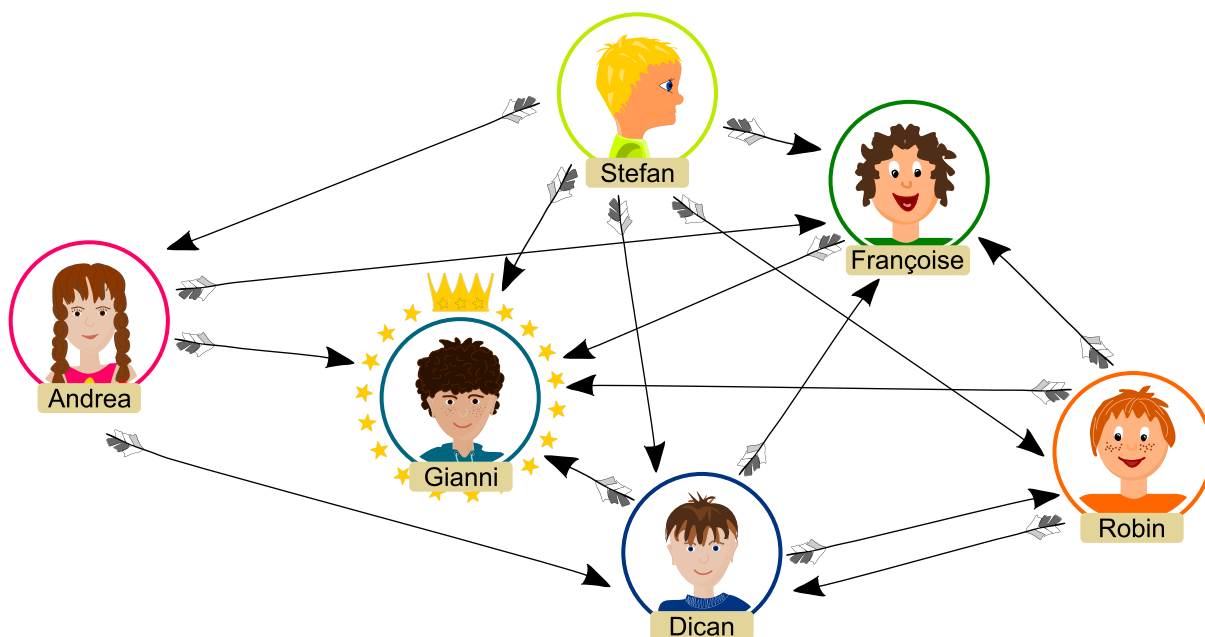
La risposta D) non può essere corretta. Non solo Françoise non è una celebrità del gruppo come descritto sopra, un gruppo può avere al massimo una sola celebrità: una celebrità non segue nessuno nel gruppo, ma tutti gli altri membri del gruppo seguono la celebrità; se ci fossero due celebrità, si dovrebbero seguire l’un l’altra, per cui non sarebbero più celebrità.

La risposta E) è anche sbagliata: il gruppo, come descritto sopra, ha Gianni come celebrità.

Questa è l’informatica!

Social networks come il fittizio TeeniGram funzionano perché i loro membri intrattengono delle (*relazioni*) tra di loro. Spesso nei social networks inizialmente queste relazioni vanno in una direzione (Andrea *segue* Dican). Naturalmente succede anche che due membri si seguono reciprocamente (Dican segue Robin e Robin segue Dican).

Queste relazioni si possono disegnare con l’aiuto di un *grafo* come nell’esempio del problema. Si usano le frecce per mostrare chi segue chi. I membri in un grafo vengono chiamati *nodi* e le frecce *archi*. Siccome gli archi hanno una direzione definita si tratta di un *grafo orientato*. Il grafo di questo problema dovrebbe assomigliare a questo:



Social networks con molti membri corrispondono spesso a grafi molto grossi. Le aziende che gestiscono questi social networks sono interessate a trovare caratteristiche particolari in questi grafi.



Una celebrità allora forse non è più qualcuno che è seguito da tutti, ma qualcuno seguito da molti. Se per esempio una celebrità in un gruppo fa della pubblicità per un certo prodotto, questa pubblicità raggiunge molti più membri rispetto a se la facesse un membro a caso. Ecco perché le celebrità cercano di attirare moltissimi *follower* e a volte usano metodi discutibili per aumentare il loro numero di follower: più follower si hanno, più è alto il profitto che si può realizzare facendo della pubblicità e del product placement (mostrare prodotti precisi). Queste persone diventano poi *influencers*, persone che influenzano altre persone.

Per gestire questo mercato, oggi i social networks più grossi usano già molti mezzi per alzare la qualità delle relazioni con i followers. A volte basta già cercare certi nomi con un browser o essere vicini a certi luoghi attraverso il riconoscimento della posizione degli smartphone, che i social network suggeriscono di “seguire” una persona o un negozio.

Parole chiave e siti web

Social networks, Grafi

- https://it.wikipedia.org/wiki/Analisi_delle_reti_sociali
- <https://it.wikipedia.org/wiki/Grafo>

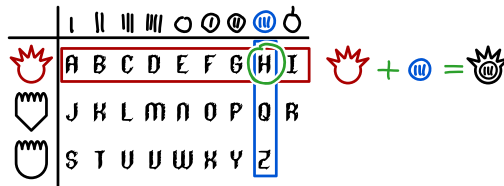




10. Messaggio dagli Antichi Castori

In fondo alla diga dei castori il castoro Dara trova un antico pezzo di legno. Nel legno sono incisi dei segni sconosciuti. Dara suppone che questa sia una tabella di un cifrario dell'epoca in cui gli Antichi Castori abitavano la diga dei castori.

Dara guarda la tabella a lungo e crede di sapere come funziona: i segni sconosciuti sono una combinazione di simboli, i quali sono indicati nelle colonne e nelle righe. La lettera "H" sarebbe cifrata in questo modo:



Dara si ricorda di aver già visto questi segni in un altro posto della diga dei castori. In effetti c'è scritto:







Cosa significa il messaggio degli Antichi Castori?



- A) SAVEWATER
- B) CLEAR DAYS
- C) SAVEMYDAM
- D) CAREFORME



Soluzione

Il primo segno è costituito dalle forme  e . Si trova quindi nella terza riga e nella prima colonna: lì c'è la lettera S. Quindi solo le risposte A) o B) possono essere corrette.

La seconda, terza e quarta lettera sono uguali nelle risposte A) e C). I segni corrispondono alle lettere A, V e E. Nella quinta lettera, tuttavia, le risposte sono di nuovo diverse. Questi segni sono costituiti dalle forme  e . Ciò corrisponde alla lettera W. Quindi la risposta corretta è la A). Anche gli ultimi quattro segni corrispondono alle lettere A, T, E e R.

C'è una scorciatoia riguardante il metodo per risolvere il problema. Se si guarda al posto del primo segno l'ultimo segno, si vede che tutte le risposte sono diverse nell'ultima lettera. Le forme  e  che costituiscono l'ultimo segno corrispondono alla lettera R e solo la risposta A) ha questa lettera alla fine.

Questa è l'informatica!

La sicurezza dei dati al giorno d'oggi è uno dei compiti più importanti della società. Uno dei metodi per proteggere i dati da letture non autorizzate è cifrarli. La scienza della cifratura delle informazioni (*crittografia*) ha almeno 3500 anni. Uno dei più antichi metodi di cifratura conosciuti si basa sulla sostituzione delle lettere con altre lettere o segni. La *cifratura* (a volte anche *codificazione*) consiste nel cifrare un *testo in chiaro* con l'aiuto di una *chiave* per farlo diventare un *testo cifrato*. La ricostruzione del testo in chiaro partendo dal testo cifrato con l'aiuto della chiave si chiama *decifrazione* (a volte anche *decodificazione*). Quando si trova il testo in chiaro dal testo cifrato senza conoscere la chiave si chiama *decriptazione*.

Il metodo di cifratura di questo problema è una cosiddetta *cifratura monoalfabetica*. Con questo procedimento per ogni lettera si sceglie esattamente un nuovo segno. Spesso vengono usati sistemi facili da ricordare. Il sistema di questo problema è simile all'alfabeto massonico. I crittoanalisti che decriptano questi testi userebbero tecniche speciali, come l'analisi delle frequenze e l'*n*-gramma nel testo cifrato, in modo da assegnare le giuste lettere decifrate ai giusti segni. Questo è generalmente possibile con la cifratura monoalfabetica, come ha mostrato Edgar Allan Poe nel suo racconto "The Gold-Bug" pubblicato nel 1843.

Cosa avrebbe potuto fare Dara, se non avesse avuto la tabella, ma avesse avuto i quattro possibili significati? Avrebbe potuto constatare che il secondo e il sesto segno, così come il quarto e l'ottavo segno sono uguali. Siccome si tratta di una cifratura monoalfabetica, deve solo trovare il testo nel quale la seconda e la sesta lettera, così come la quarta e l'ottava lettera sono uguali, e questo corrisponde solo alla risposta A).

Parole chiave e siti web

Crittografia, Cifratura monoalfabetica

- <https://it.wikipedia.org/wiki/Crittografia>
- https://it.wikipedia.org/wiki/Cifrario_a_sostituzione
- https://it.wikipedia.org/wiki/Cifrario_pigpen
- https://it.wikipedia.org/wiki/Analisi_delle_frequenze
- <https://it.wikipedia.org/wiki/N-gramma>
- https://it.wikipedia.org/wiki/Lo_scarabeo_d'oro



- <http://users.telenet.be/d.rijmenants/en/goldbug.htm>





11. Caratteri cinesi variopinti

La struttura dei caratteri cinesi ci appare estranea. Per capire meglio la composizione di alcuni caratteri cinesi si può pensare a questo schema, nel quale si distinguono cinque parti, sopra ▲, sotto ⊙, sinistra ◻, destra ● e centro ☆:



Queste parti possono essere disposte come quattro strutture:

Struttura	sinistra- centro-destra	sinistra- destra	sopra- centro-sotto	sopra- sotto
Esempio di carattere	川	儿	三	昌
Esempio di analisi				

Quale analisi mostra la disposizione corretta secondo lo schema dei tre caratteri cinesi 劳, 二, e 八?

- A)
- B)
- C)
- D)

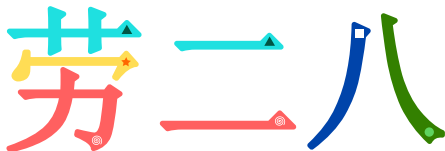


Soluzione

Il primo carattere 劳 corrisponde alla struttura sopra-centro-sotto, quindi il tratto superiore è azzurro ▲, il tratto centrale è giallo ★ e il tratto inferiore è rosa ⊗.

Il secondo carattere 二 corrisponde alla struttura sopra-sotto, quindi il tratto superiore è azzurro ▲ e quello inferiore è rosa ⊗.

Il terzo carattere 八 corrisponde alla struttura sinistra-destra, quindi il tratto a sinistra è blu scuro ■ e il tratto a destra è verde ■.



Quindi la risposta corretta è la B)

Nella risposta A) il secondo carattere 二 viene analizzato correttamente, ma ad entrambi i caratteri 劳 e 八 vengono assegnati dei colori sbagliati: per 劳 il colore superiore è sbagliato, per 八 i due colori sono scambiati.

Nella risposta C) tutti i caratteri sono analizzati erroneamente. Nel primo carattere il colore centrale e quello inferiore sono stati scelti erroneamente, nel secondo carattere il colore superiore è stato scelto erroneamente e per il terzo carattere entrambi i colori sono stati scelti erroneamente.

Nella risposta D) il carattere 八 è stato analizzato correttamente, ma per 劳 il colore superiore e quello inferiore sono sbagliati e per 二 entrambi i colori sono stati scelti erroneamente.

Questa è l'informatica!

La scrittura cinese è composta da complessi caratteri composti. Anche nelle varianti semplificate ci sono oltre 200 elementi di base (*radicali*), dai quali vengono assemblati i caratteri. Questi vengono scritti l'uno a fianco all'altro o uno sotto l'altro, in modo da formare effettivamente delle strutture come spiegato nel problema. In questo modo possono essere combinati migliaia di caratteri diversi. Se si devono imparare questi caratteri bisogna imparare la loro composizione. Per farlo vengono spesso usati dei colori. L'alfabeto latino usato da noi funziona diversamente: una *lettera* corrisponde ad un suono (con eccezioni come sc seguito da "i" o "e", che viene pronunciato [ʃ] e non [khe]).

Cosa ha a che fare questo con l'informatica? Da un lato questi segni devono poter essere rappresentati da un computer. Per questo ci sono diversi approcci, un approccio sfrutta i radicali descritti in questo problema. Dall'altro bisogna essere in grado di poter cercare delle parole, ad esempio nei dizionari o nelle enciclopedie. I radicali più comunemente usati al giorno d'oggi provengono da un dizionario elaborato dal 1710 al 1716 sotto l'imperatore Kangxi. È ordinato secondo il numero di tratti in ogni radicale.

Parole chiave e siti web

Caratteri cinesi

- [https://it.wikipedia.org/wiki/Radicali_\(cinese\)](https://it.wikipedia.org/wiki/Radicali_(cinese))
- https://it.wikipedia.org/wiki/Caratteri_cinesi_semplificati
- https://en.wikipedia.org/wiki/Chinese_character_encoding
- https://en.wikipedia.org/wiki/Chinese_input_methods_for_computers
- https://it.wikipedia.org/wiki/Dizionario_di_Kangxi



- https://it.wikipedia.org/wiki/Alfabeto_latino
- https://it.wikipedia.org/wiki/Digrammi_e_trigrammi_della_lingua_italiana

Le lettere cinesi sono:





- 川: https://en.wikipedia.org/wiki/Radical_47
- 儿: https://en.wikipedia.org/wiki/Radical_10
- 吕: [https://en.wikipedia.org/wiki/L%C3%BC_\(surname\)](https://en.wikipedia.org/wiki/L%C3%BC_(surname))
- 二: https://en.wikipedia.org/wiki/Radical_7
- 三: <https://en.wikipedia.org/wiki/3>
- 八: https://en.wikipedia.org/wiki/Radical_12
- 劳: <https://en.wiktionary.org/wiki/%E5%8A%B3>



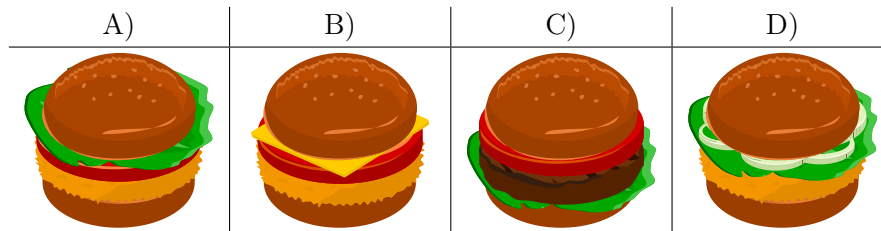


12. Ingredienti degli hamburger

BeaverBurger offre sei ingredienti (A, B, C, D, E e F) per i suoi hamburger fatti in casa. La tabella seguente mostra gli ingredienti per i quattro esempi di hamburger, dove gli ingredienti non sono ordinati per forza come negli esempi di hamburger:

Burger				
Ingredienti	C, F	A, B, E	B, E, F	B, C, D







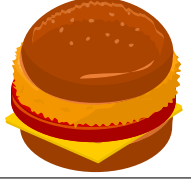

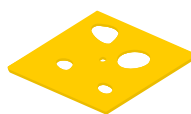


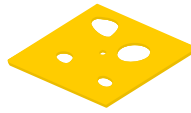

Quale hamburger contiene gli ingredienti A, E e F?









Soluzione



Per scoprire quale ingrediente è assegnato a quale lettera bisogna sempre confrontare due hamburger uno con l'altro:

Hamburger confrontati		Lettera in comune	Ingrediente in comune
		F	
		C	
		B	
		B (appena identificato)	
		E	

Due ingredienti si trovano solo in un hamburger. Siccome conosciamo già tutte le altre lettere possiamo quindi identificare gli ingredienti corrispondenti:

Hamurger particolare	Lettera particolare	Ingrediente particolare
	A	
	D	

Quindi l'hamburger che cerchiamo con gli ingredienti A, E e F, composto dagli ingredienti , 

e  è l'hamburger della risposta A) 



Questa è l'informatica!

L'*inferenza logica* è la base per molti ragionamenti, anche nell'informatica. Per risolvere questo problema bisogna applicarla in modo intensivo: attraverso il confronto degli hamburger con gli stessi ingredienti si possono dedurre informazioni che prima erano sconosciute (quale ingrediente corrisponde a quale lettera).

Nel caso di questo problema gli ingredienti in comune di due hamburger corrispondono all'*intersezione* degli ingredienti dei due hamburger. Contiene solo gli ingredienti che sono contenuti in entrambi gli hamburger. Per il primo confronto si scriverebbe quindi $\{C, F\} \cap \{B, E, F\} = \{F\}$. Il contrario dell'intersezione sarebbe d'altronde l'*unione* $\{C, F\} \cup \{B, E, F\} = \{B, C, E, F\}$, contiene tutti gli elementi che sono contenuti almeno in un hamburger.

Per scoprire gli ingredienti che sono contenuti solo in un hamburger si può usare la *differenza*. Contiene solo gli ingredienti del primo insieme che non sono contenuti nel secondo insieme. Per il primo hamburger particolare si potrebbe ad esempio scrivere: $\{A, B, E\} \setminus (\{C, F\} \cup \{B, E, F\} \cup \{B, C, D\}) = \{A, B, E\} \setminus \{B, C, D, E, F\} = \{A\}$.

La teoria degli insiemi si conosce magari dalle lezioni di matematica. Nell'informatica viene usata ad esempio nelle banche dati. Ma si può anche convertire la teoria degli insiemi 1 : 1 in logica, chiamata anche algebra di Boole, e anche essa viene usata in molti ambiti dell'informatica.

Parole chiave e siti web

Inferenza logica, Teoria degli insiemi, Logica

- <https://it.wikipedia.org/wiki/Inferenza>
- <https://it.wikipedia.org/wiki/Insieme>
- https://it.wikipedia.org/wiki/Algebra_di_Boole



A. Autori dei quesiti

 Tony René Andersen	 Alisher Ikramov	 Wolfgang Pohl
 Michelle Barnett	 Thomas Ioannou	 Nol Premasathian
 Michael Barot	 Felipe Jiménez	 J.P. Pretti
 Linda Bergsveinsdóttir	 Anna Laura John	 Andrea Schrijvers
 Daniela Bezáková	 Mile Jovanov	 Vipul Shah
 Laura Braun	 Adem Khachnaoui	 Taras Shpot
 Mony Chanroath	 Injoo Kim	 Jacqueline Staub
 Marios Choudary	 Jihye Kim	 Nikolaos Stratis
 Anton Chukhnov	 Vaidotas Kinčius	 Gabrielė Stupurienė
 Andrew Csizmadia	 Jia-Ling Koh	 Maciej M. Sysło
 Valentina Dagienė	 Sophie Koh	 Bundit Thanasopon
 Christian Datzko	 Dennis Komm	 Monika Tomcsányiová
 Susanne Datzko	 Anja Koron	 Peter Tomcsányi
 Lanping Deng	 Bohdan Kudrenko	 Jiří Vaníček
 Marissa Engels	 Regula Lacher	 Ela Veza
 Olivier Ens	 Inggriani Liem	 Florentina Voboril
 Sonali Gogate	 Karolína Mayerová	 Jing-Jing Yang
 Martin Guggisberg	 Anna Morpurgo	 Xing Yang
 Vernon Gutierrez	 Tom Naughton	 Khairul A. Mohamad Zaki
 Juraj Hromkovič	 Pia Niemelä	



B. Sponsoring: concorso 2019

HASLERSTIFTUNG

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

ROBOROBO

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

**bischof
berger**

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

verkehrshaus.ch

<http://www.verkehrshaus.ch/>
Musée des transports, Lucerne

 **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 (Musée des transports, Lucerne)

 **UBS**

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

bbv
Software Services

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

PRESENTEX
Das Geschenk - die gute Werbung

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

OXOCARD

<http://www.oxocard.ch/>
OXOcard
OXON

 **DIARTIS**

<http://www.diartis.ch/>
Diartis AG



<https://educatec.ch/>
educaTEC



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



AUSBILDUNGS- UND BERATUNGSZENTRUM
FÜR INFORMATIKUNTERRICHT

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



<http://www.hepl.ch/>
Haute école pédagogique du canton de Vaud



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



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

Scuola universitaria professionale
della Svizzera italiana

<http://www.supsi.ch/home/supsi.html>
La Scuola universitaria professionale della Svizzera italiana (SUPSI)



<https://www.zhdk.ch/>
Zürcher Hochschule der Künste



C. Ulteriori offerte

010100110101011001001001
010000010010110101010011
010100110100100101000101
001011010101001101010011
01001001010010010010001

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.