



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

Quesiti e soluzioni 2022

3^o e 4^o anno scolastico

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

A cura di:

Susanne Datzko, Nora A. Escherle, Masiar Babazadeh,
 Christian Giang, Jean-Philippe Pellet

010100110101011001001001
 010000010010110101010011
 010100110100100101000101
 001011010101001101010011
 010010010100100100100001

SS! I

www.svia-ssie-ssii.ch
 schweizerischerverein für informatik in
 1erausbildung // société suisse pour l'infor
 matique dans l'enseignement // società sviz
 zera per l'informatica nell'insegnamento



Hanno collaborato al Castoro Informatico 2022

Masiar Babazadeh, Susanne Datzko, Jean-Philippe Pellet, Giovanni Serafini, Bernadette Spieler

Capo progetto: Nora A. Escherle

Un particolare ringraziamento per il lavoro sui quesiti del concorso Svizzero va a:

Juraj Hromkovič, Christian Datzko, Jens Gallenbacher, Regula Lacher: ETH Zurich, Ausbildungs- und Beratungszentrum für Informatikunterricht

Tobias Berner: Pädagogische Hochschule Zürich

Waël Almoman: Collège Voltaire

La scelta dei quesiti è stata svolta in collaborazione con gli organizzatori dei concorsi in Germania, Austria, Ungheria, Slovacchia e Lituania. Ringraziamo specialmente:

Valentina Dagienė, Tomas Šiaulyš, Vaidotas Kinčius: Bebras.org

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

Wilfried Baumann, Liam Baumann, Anoki Eischer, Thomas Galler, Benjamin Hirsch, Martin Kandlhofer, Katharina Resch-Schobel: Österreichische Computer Gesellschaft

Gerald Futschek, Florentina Voboril: Technische Universität Wien

Zsuzsa Pluhár: ELTE Informatikai Kar, Ungheria

Michal Winzcer: Comenius University, Slovacchia

La versione online del concorso è stata creata su cuttle.org. Ringraziamo per la buona collaborazione: Eljakim Schrijvers, Justina Dauksaite, Dave Oostendorp, Alieke Stijf, Kyra Willekes, Jo-Ann Bolten: cuttle.org, Olanda

Chris Roffey: UK Bebras Administrator, Regno Unito

Per il supporto durante le settimane del concorso ringraziamo:

Hanspeter Erni: Direttore scuola media di Rickenbach

Christoph Frei: Chragokyberneticks (Logo Informatik-Biber Schweiz)

Dr. Andrea Leu, Maggie Winter, Lena Frölich: Senarclens Leu + Partner AG

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 Christian Giang.



INFORMATIK-BIBER SCHWEIZ
CASTOR INFORMATIQUE SUISSE
CASTORO INFORMATICO SVIZZERA

Il Castoro Informatico 2022 è stato organizzato dalla Società Svizzera per l'Informatica nell'Insegnamento (SSII) con il sostegno determinante della fondazione Hasler. Gli sponsor del concorso sono l'Ufficio per l'economia e il lavoro del Cantone di Zurigo e UBS.

Questo quaderno è stato creato il 22 novembre 2023 con il sistema per la preparazione di testi \LaTeX . Ringraziamo Christian Datzko per lo sviluppo del sistema di generazione dei testi che ha permesso di generare le 36 versioni di questa brochure (divise per lingua e livello scolastico). Il sistema è stato riprogrammato basandosi sul sistema precedente, sviluppato nel 2014 assieme a Ivo Blöchliger. Ringraziamo Jean-Philippe Pellet per lo sviluppo del sistema `bebras`, utilizzato dal 2020 per la conversione dei documenti sorgente dai formati Markdown e YAML.

Nota: Tutti i link sono stati verificati l'01.12.2022.



I quesiti sono distribuiti con Licenza Creative Commons Attribuzione – Non commerciale – Condividi allo stesso modo 4.0 Internazionale. Gli autori sono elencati a pagina 47.



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.

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 2022 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

Ogni categoria aveva quesiti classificati in tre livelli di difficoltà: facile, medio e difficile. 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 inizia 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. Questi quesiti presentavano livelli di difficoltà diversi nei vari gruppi di età.

Alcuni quesiti sono indicati come «bonus» per determinate categorie di età: non contano nel totale dei punti, ma vengono utilizzati come spareggio per punteggi identici in caso di qualificazione agli eventuali turni successivi.

Per ulteriori informazioni:

SVIA-SSIE-SSII Società Svizzera per l'Informatica nell'Insegnamento
Castoro Informatico
Masiar Babazadeh

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

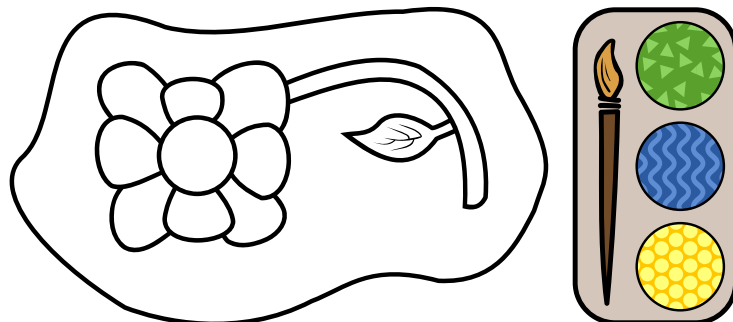


Indice

Hanno collaborato al Castoro Informatico 2022	i
Premessa	iii
Indice	v
1. Colorazione	1
2. Caramelle preferite	5
3. Biblioteca	7
4. Alveare	11
5. Permutazioni	15
6. Tartaruga e lepre	19
7. Piramide colorata	23
8. Ricetta hamburger	27
9. Collana da marinaio	31
10. Cuore composto	35
11. Bulloni e dadi	39
12. I vicini di Lili	43
A. Autori dei quesiti	47
B. Sponsoring: concorso 2022	48
C. Ulteriori offerte	49



1. Colorazione

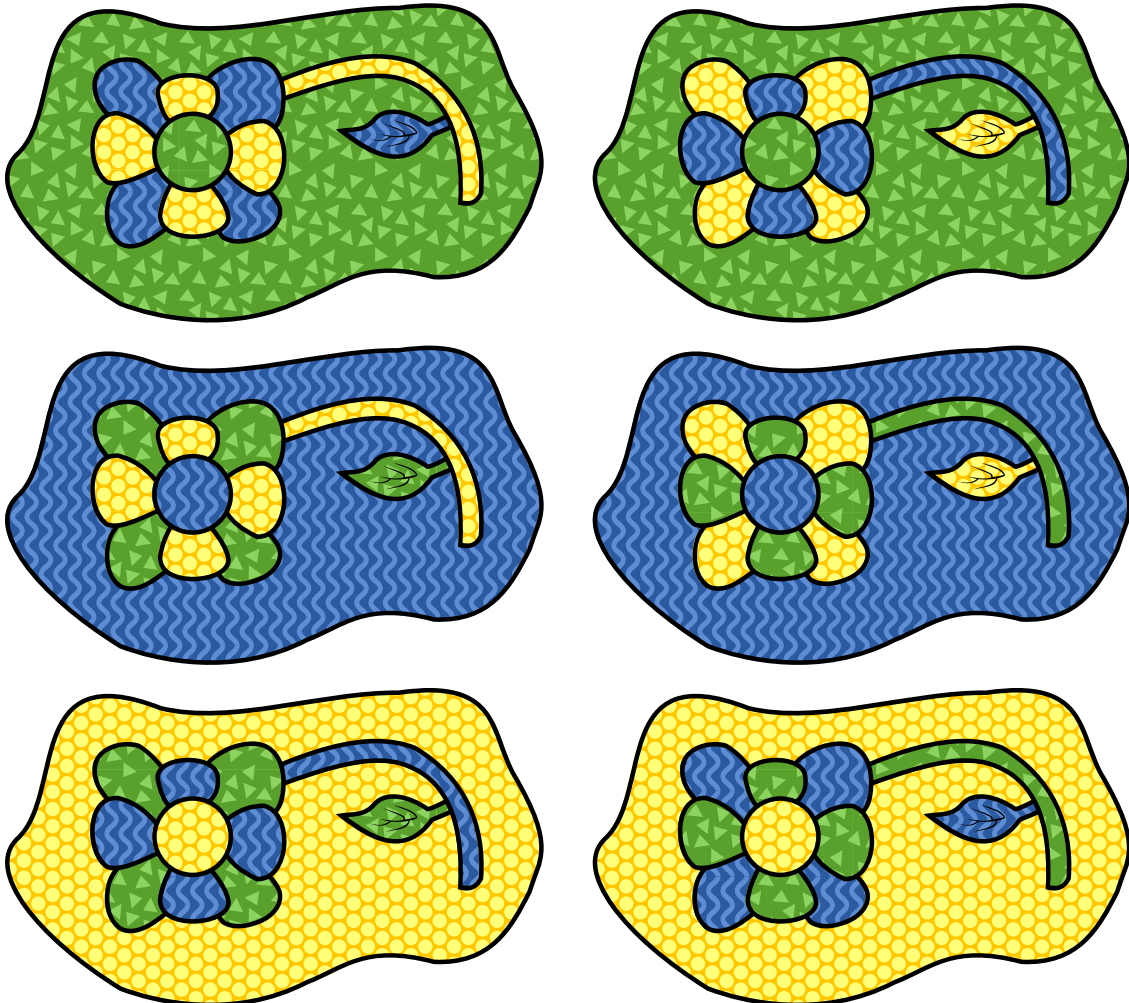


Colora l'immagine di verde, giallo e blu in modo che non ci siano due aree dello stesso colore che si toccano.



Soluzione

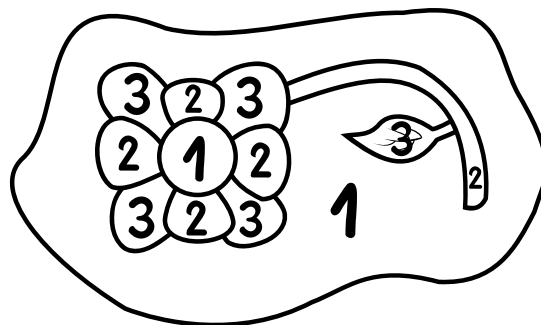
Qui si possono vedere tutte le possibilità di come colorare l'immagine.



Come si può trovare una delle soluzioni?

Si sceglie innanzitutto un colore per la superficie più esterna, ad esempio il giallo. La maggior parte delle altre superfici tocca la superficie gialla. Quindi devono essere colorati di blu e verde. Si inizia con una di queste aree e si alternano i colori. Il centro del fiore può essere quindi giallo.

Questa immagine numerata mostra questa strategia indipendentemente dai colori:





Questa è l'informatica!

Questo compito consiste nell'assegnare i colori alle aree, rispettando determinati vincoli. In informatica e in matematica, tali problemi sono noti come *colorazione di grafi*.

I problemi di colorazione di grafi hanno molte applicazioni e spesso è importante utilizzare il minor numero possibile di colori. Tra gli esempi ci sono la divisione delle squadre in un torneo sportivo, la suddivisione delle persone in gruppi o l'assegnazione delle frequenze delle stazioni radio. Nelle applicazioni dei problemi di colorazione di grafi, di solito ci sono più superfici che in questo compito. Spesso non è più possibile trovare una colorazione con pochi colori a mano. Gli informatici hanno bisogno di un computer per risolvere questi compiti.

Il problema della colorazione di grafi può essere utilizzato anche per le mappe. In questo caso, l'obiettivo è trovare una colorazione in modo che i paesi vicini non siano colorati allo stesso modo. Per ogni mappa è possibile trovare una tale colorazione con solo quattro colori ma non è facile da dimostrare. Solo nel 1976 i matematici Kenneth Appel e Wolfgang Haken ci riuscirono. Per farlo, hanno utilizzato i computer per verificare un gran numero di eccezioni e controesempi. Altre persone non possono riprodurlo senza computer, ecco perché ci sono altri matematici che criticano l'uso dei computer in questa prova o nelle prove in generale.

Parole chiave e siti web

- Colorazione di grafi: https://it.wikipedia.org/wiki/Colorazione_dei_grafi
- Teorema dei quattro colori:
https://it.wikipedia.org/wiki/Teorema_dei_quattro_colori





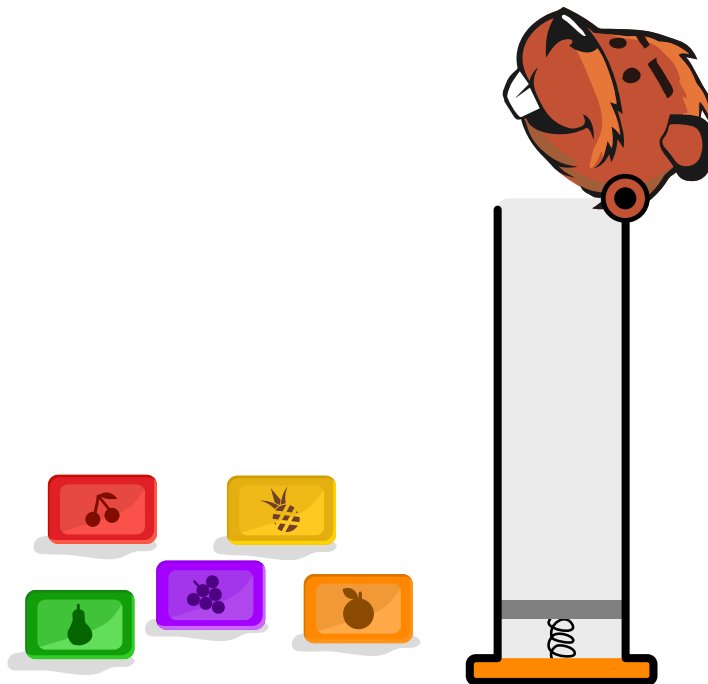
2. Caramelle preferite

Anna riempie un distributore con cinque caramelle. Poi potrà mangiare le caramelle una dopo l'altra mentre escono dalla parte superiore del distributore.

Vuole mangiare le caramelle una dopo l'altra in quest'ordine:



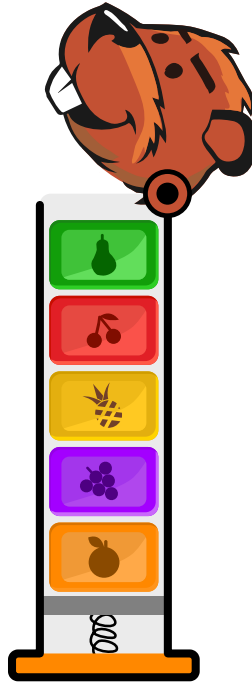
Come deve fare Anna per riempire le caramelle nel distributore?





Soluzione

Affinché le caramelle escano dal distributore nell'ordine corretto, è importante capire che la caramella che viene inserita per prima esce per ultima. Ciò significa che il distributore deve essere riempito nel seguente modo:



Questa è l'informatica!

Se Anna si limita a inserire le caramelle nel distributore nell'ordine in cui vuole mangiarle, queste usciranno esattamente nell'ordine opposto. Pertanto, Anna decide prima l'ordine che desidera! Poi immagina come deve riempire il distributore per ottenere i dolci nell'ordine desiderato.

Per gli informatici è spesso importante pensare anche all'ordine. L'ordine utilizzato in questo compito è chiamato «ordine di pila». Le *pila* sono strutture di memoria che aggiungono e tolgono oggetti in un certo ordine: Lavorano secondo il principio «*Last in First out*» (*LIFO*). Ciò significa che gli oggetti aggiunti per ultimi vengono eliminati per primi.

Parole chiave e siti web

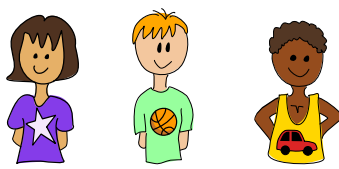
- Pila: [https://it.wikipedia.org/wiki/Pila_\(informatica\)](https://it.wikipedia.org/wiki/Pila_(informatica))
- Last in First out (LIFO): <https://it.wikipedia.org/wiki/LIFO>



















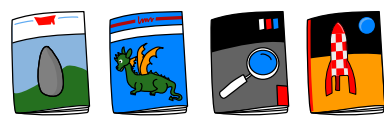
3. Biblioteca

Dei bambini prendono in prestito alcuni libri dalla biblioteca. La biblioteca scrive in una tabella chi ha preso in prestito quale libro.

Quale libro hanno preso in prestito più spesso i bambini?







Soluzione



La risposta corretta è C):

Quanto segue è corretto:

- Tre bambini hanno preso in prestito il libro con il razzo.
- Un bambino ha preso in prestito il libro con la lente d'ingrandimento.
- Due bambini hanno preso in prestito il libro con il drago.
- Un bambino ha preso in prestito il libro con il menhir.

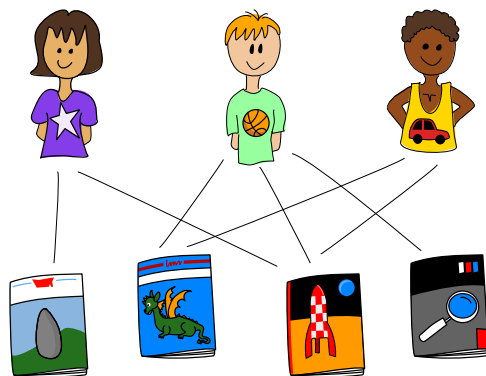


Quindi è il libro con il razzo che è stato preso in prestito più spesso.

Questa è l'informatica!

È fantastico che ai bambini del Concorso Castoro Informatica piaccia leggere libri!

Ma abbiamo davvero bisogno di un tavolo con bambini e libri per rappresentare i desideri dei bambini? Non funzionerebbe anche disegnare semplicemente delle linee?





Sarebbe più facile per gli esseri umani, ma non per i computer. I computer non sono bravi a leggere le righe. Ma sono molto bravi a lavorare con i tavoli. Se vogliamo che i computer ci aiutino a capire, ad esempio, quale bambino ha preso in prestito un libro o quale persona possiede un conto in banca, di solito è una buona idea mostrarlo in tabelle.

Le tabelle sono state introdotte 4000 anni fa a Babilonia per memorizzare informazioni sulle *relazioni*. Questa capacità di memorizzare relazioni rende le tabelle un importante concetto di base dei database relazionali.

Le tabelle rappresentano le relazioni tra le cose (o le persone). Le relazioni determinano il modo in cui rappresentare le informazioni nelle tabelle. Ad esempio, se la regola fosse che ogni bambino può prendere in prestito un solo libro, la tabella avrebbe una sola riga per ogni bambino. Nel nostro esempio della biblioteca, è giusto che i bambini possano prendere in prestito diversi libri, e che possano persino prendere in prestito gli stessi libri degli altri bambini. In questo caso, abbiamo bisogno di una tabella speciale che colleghi i bambini e i libri e che possa elencare lo stesso bambino più volte e anche lo stesso libro più volte.

Il tavolo di circolazione è pratico. Se manca un libro, ad esempio, il bibliotecario può verificare se è stato prestato. La tabella di circolazione ha due colonne e molte righe. Nella prima colonna viene inserito il bambino che prende in prestito un libro e nella seconda colonna il libro. In questo modo, alla domanda su quale libro sia stato prestato di più si può rispondere semplicemente contando il numero nella seconda colonna.

Questo compito potrebbe essere svolto anche da un computer. Se si tratta di una grande biblioteca con molte migliaia di libri, non c'è altro modo! In una biblioteca così grande, non solo il tavolo di circolazione verrebbe mantenuto. Ci sarà anche un archivio clienti (tabella clienti) in cui saranno memorizzate tutte le informazioni sui clienti, come nome, indirizzo e numero di telefono e un indice dei libri (tabella libri) con informazioni sui libri, come autore e titolo. In questo modo, la tabella di circolazione rimane snella perché contiene solo le relazioni (cioè chi ha preso in prestito quale libro) tra i clienti e i libri.

In informatica, tali tabelle sono chiamate *basi di dati* relazionali.

Parole chiave e siti web

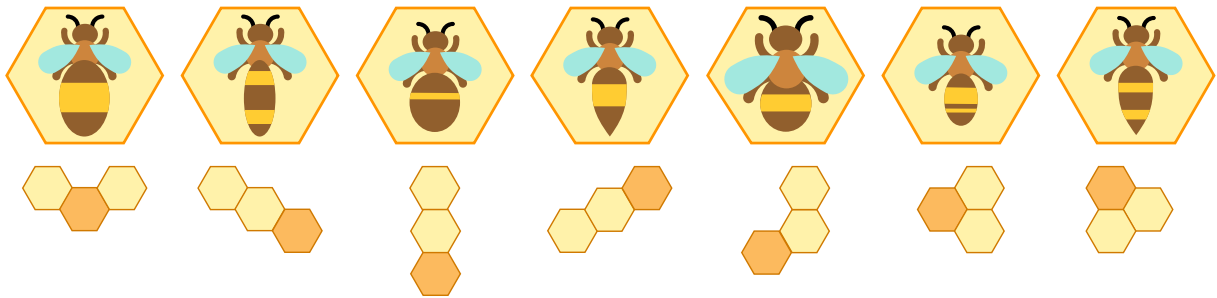
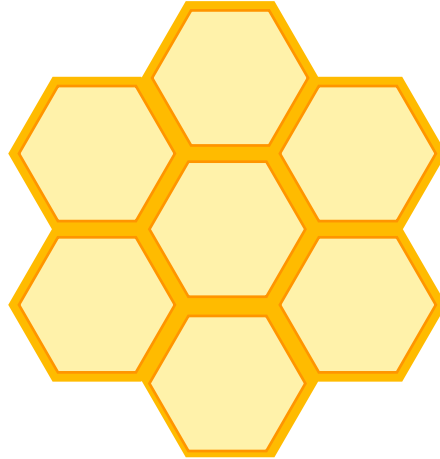
- Base di dati («*database*»): https://it.wikipedia.org/wiki/Base_di_dati





4. Alveare

Un castoro ha bisogno di aiuto per alimentare tutte le api del suo alveare.



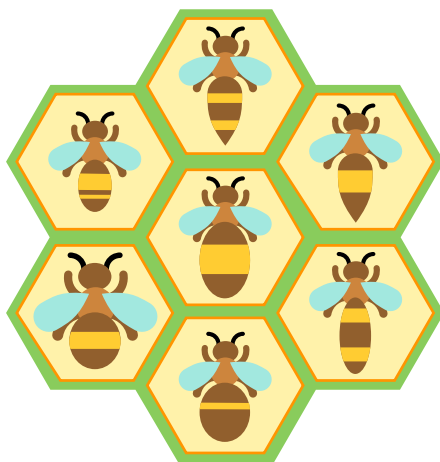
Sotto ogni ape, una regola indica in quale cella può essere alimentata l'ape.

Alimenta le api nell'alveare. Rispetta le regole sotto le api.

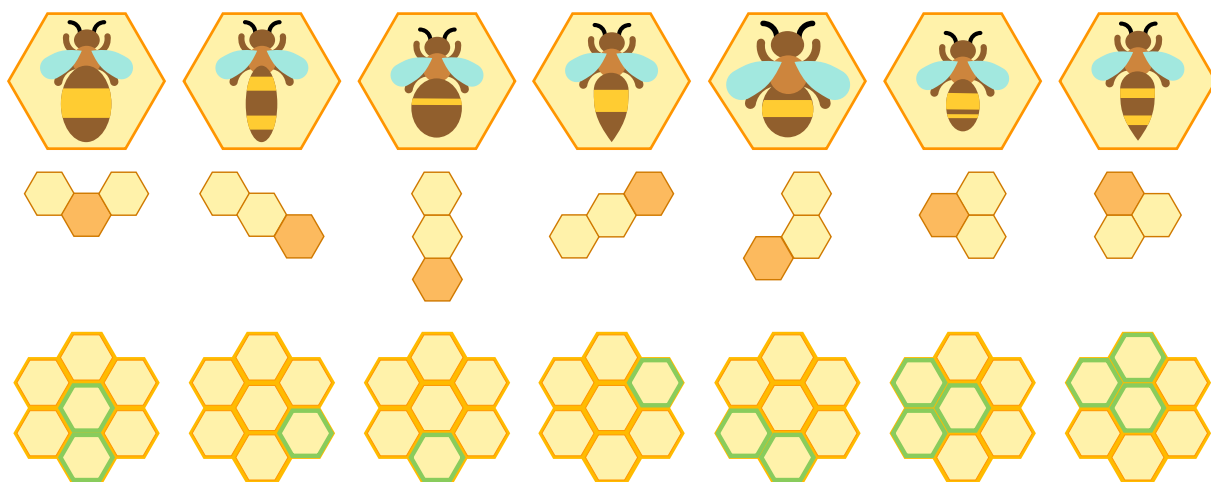


Soluzione

Le api possono essere alimentate nell'alveare solo in questo modo:



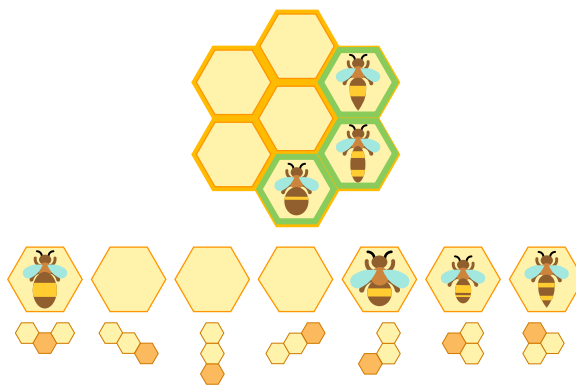
Il compito può essere risolto per tentativi ed errori. Ma questo può richiedere molto tempo. Per trovare un modo più veloce, si può osservare più da vicino le regole delle api. Nella figura seguente è possibile vedere ogni ape e la relativa regola. Le celle in cui l'ape può essere alimentata secondo la sua regola sono delineate in verde.



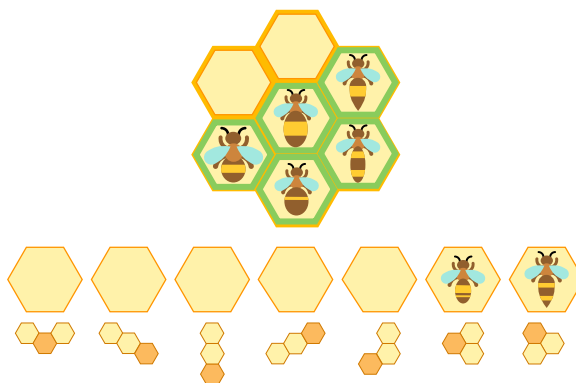
Si può notare che per alcune api ci sono più possibilità di alimentarle secondo le regole, per altre ce ne sono meno. Per tre api c'è solo un modo per alimentarle.

Per risolvere il compito in modo più rapido rispetto ai tentativi, si può procedere come segue:

Per prima cosa alimentare le api per le quali è possibile utilizzare una sola cella dell'alveare.



Allora c'è solo una cella possibile per le due api successive.



Allo stesso modo, ci si occupa delle ultime due api.

Questa è l'informatica!

In questo compito, sette api devono essere alimentate in sette celle diverse. Ci sono molti modi per prendersi cura delle api. Se si tiene conto delle regole, il numero di possibilità si riduce già di molto, ma è ancora così alto che provare tutte le possibilità richiederebbe uno sforzo considerevole. La chiave per una rapida soluzione del compito è l'ordine giusto. In questo caso, abbiamo iniziato con gli elementi più ristretti, cioè le api che hanno una sola cella possibile, per limitare il numero di casi da studiare.

Un approccio risolutivo di questo tipo è chiamato *euristico* in informatica. Grazie a un'ingegnosa sequenza di soluzioni, la soluzione esatta può essere trovata con pochi passaggi. Tuttavia, per alcuni problemi, come la pianificazione di un percorso tra diverse località in un sistema di navigazione, un approccio euristico va a scapito della precisione. Questo perché le soluzioni possibili sono numerosissime. Per avere la garanzia di trovare il percorso migliore, bisognerebbe calcolare e confrontare tutti i percorsi possibili attraverso l'intera rete di percorsi, il che comporterebbe un'enorme quantità di calcoli. Provando prima le possibilità che hanno maggiori probabilità di portare a una buona soluzione, lo sforzo computazionale può essere ridotto in modo significativo. In questo modo, è possibile determinare un buon percorso in pochi secondi invece di calcolare il migliore in anni.



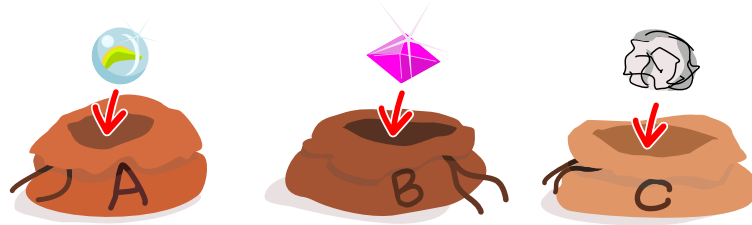
Parole chiave e siti web

- Euristicia: <https://it.wikipedia.org/wiki/Euristica>
- Problema del commesso viaggiatore:
https://it.wikipedia.org/wiki/Problema_del_commesso_viaggiatore

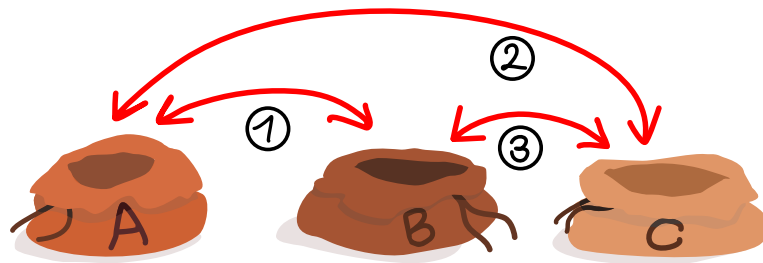


5. Permutazioni

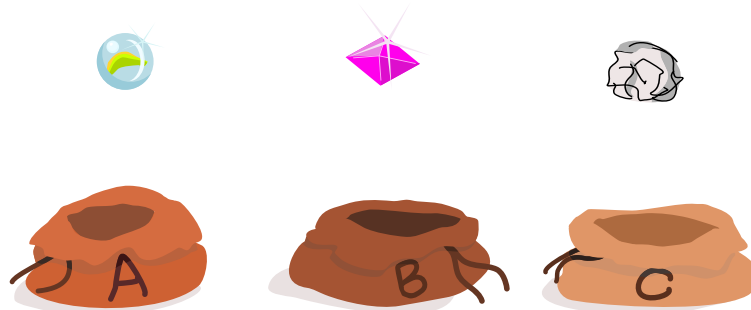
Lila mette una biglia nel sacchetto A, una pietra preziosa nel sacchetto B e un pezzo di carta nel sacchetto C.



Poi scambia il contenuto del sacchetto A e del sacchetto B, quindi il contenuto di A e C e infine scambia il contenuto di B e C.



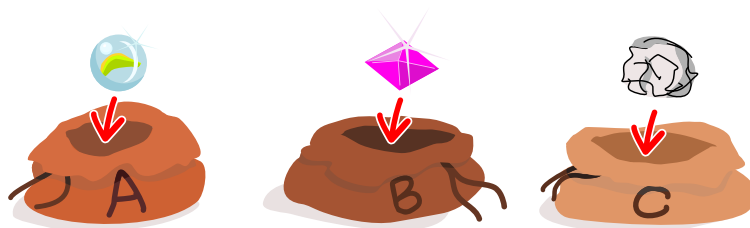
Dove sono i tre oggetti?





Soluzione

All'inizio abbiamo questa disposizione delle 3 cose nei sacchetti:



Lila scambia le cose 3 volte. Dopo il primo scambio (A-B) i sacchetti hanno questa disposizione:



Dopo il secondo scambio (A-C):



Dopo il terzo e ultimo scambio (B-C):



Pertanto, alla fine, il pezzo di carta si trova in A, la pietra preziosa in B e la biglia in C. Questo risultato si sarebbe potuto ottenere anche in modo più semplice, cioè con un unico scambio dei contenuti di A e C.

Questa è l'informatica!

Qui si tratta di sequenzialità di cose. Questa sequenza di cose viene anche chiamata «disposizione». Un ordine diverso rappresenta una disposizione diversa. Uno scambio cambia l'ordine delle cose e quindi porta a una disposizione diversa. Nel nostro compito, abbiamo la disposizione biglia-pietra-carta all'inizio e la disposizione carta-pietra-biglia dopo le 3 permutazioni.

Una domanda interessante è quante disposizioni diverse possono avere 3 cose. Possiamo semplificare un po' la situazione per ora e fare tutte le disposizioni, eseguendo solo un'azione. Per le altre due cose ci sono solo due disposizioni. Se la biglia si trova al primo posto, le due disposizioni sono:



Biglia-pietra-carta

Biglia-carta-pietra

Pertanto, anche per gli altri due oggetti esistono solo due disposizioni diverse. Quindi ci sono altre 4 disposizioni delle 3 cose:

Pietra-biglia-carta

Pietra-carta-biglia

Carta-biglia-pietra

Carta-pietra-biglia

È inoltre interessante notare che è possibile creare qualsiasi disposizione solo con le permutazioni. Ciò richiede al massimo $n - 1$ permutazioni per n cose.

Parole chiave e siti web

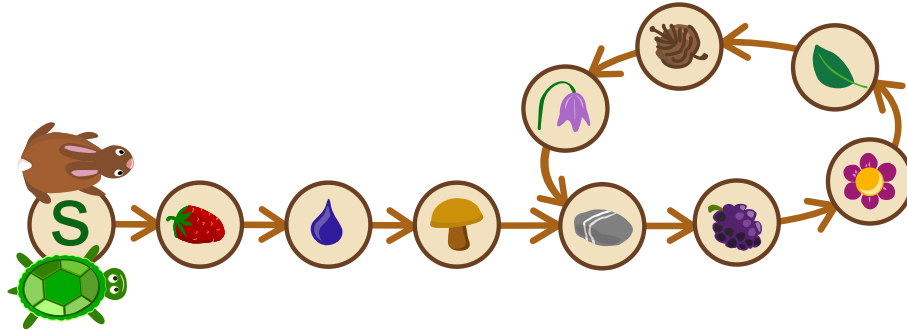
- Permutazione: <https://it.wikipedia.org/wiki/Permutazione>





6. Tartaruga e lepre

Una tartaruga 🐢 e una lepre 🐇 stanno facendo una gara. Utilizzano questa pista.



Iniziano nello stesso momento sul campo di partenza. Vanno di campo in campo e seguono le frecce.

In un minuto, ...

- ... la tartaruga avanza di un campo.
- ... la lepre avanza di due campi.

In quale campo la tartaruga e la lepre si incontrano per la prima volta dopo la partenza?

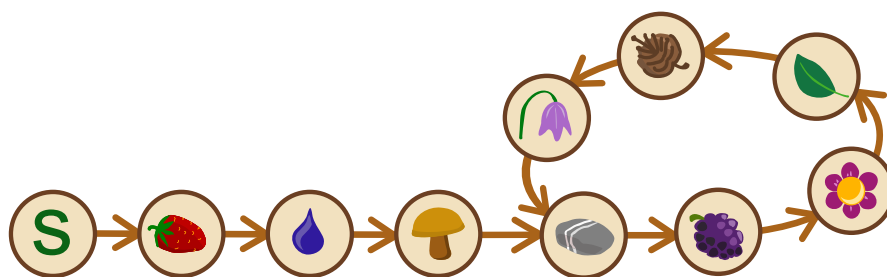


Soluzione

Tartaruga e lepre si incontrano per la prima volta sul campo 🌸. Puoi seguirlo facilmente con due dita.

La seguente tabella mostra i campi della tartaruga e della lepre per ogni minuto:

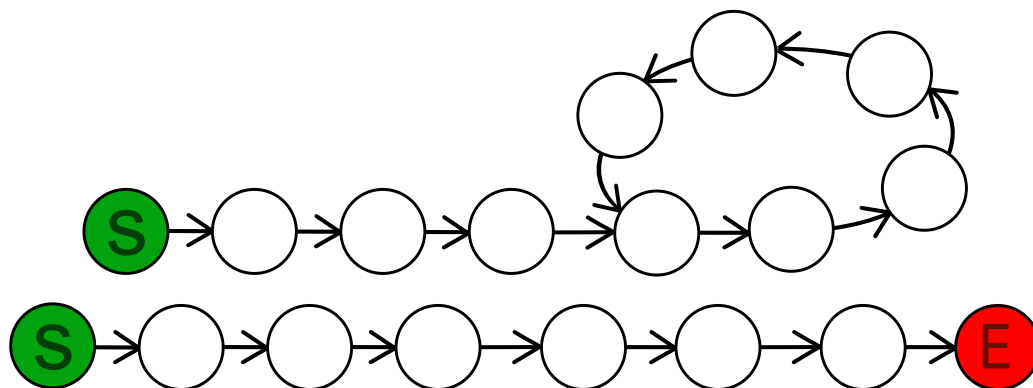
Minuti dopo l'inizio	0	1	2	3	4	5	6	7	8	9	10	11	12	13	...
	S	🍓	💧	🍄	🪨	🍇	🌸	🌿	🌰	🌺	🪨	🍇	🌸	🌿	...
	S	💧	🪨	🌸	🌰	🪨	🌸	🌰	🪨	🌸	🌰	🪨	🌸	🌰	...



Questa è l'informatica!

In questo compito, la gara si svolge su una pista speciale. È composta da singoli campi e da frecce che indicano il campo successivo. La particolarità è che la pista termina con un cerchio in cui i corridori possono correre all'infinito. La tartaruga e la lepre possono incontrarsi in questo compito solo perché questi 6 campi formano un cerchio o un *ciclo*.

In informatica, una traccia di corsa come quella descritta nel compito verrebbe chiamata *lista*. Un cerchio di campi che si riferiscono l'uno all'altro come nel compito si chiama *ciclo*. In una lista, ogni *vertice* si riferisce al massimo a un altro vertice. Esistono liste con un ciclo, come in questo compito, e liste senza ciclo.



Se una lista non ha cicli, allora la lista consiste in una catena lineare di vertici. Allora deve esistere anche un campo finale dal quale non parte alcuna freccia. Il famoso informatico Robert W. Floyd



(1936–2001) ha ideato un algoritmo in grado di distinguere facilmente se una lista ha un ciclo o consiste in una catena lineare. In modo simile al nostro compito, lascia che la lepre e la tartaruga inizino a correre nel campo di partenza. Se la tartaruga e la lepre arrivano nello stesso campo nello stesso momento, c'è un ciclo. Nel momento in cui la lepre raggiunge il campo finale o il campo precedente, non c'è più nessun ciclo e l'algoritmo è terminato.

Parole chiave e siti web

- Lista concatenata: https://it.wikipedia.org/wiki/Lista_concatenata
- Vertice: [https://it.wikipedia.org/wiki/Vertice_\(teoria_dei_grafi\)](https://it.wikipedia.org/wiki/Vertice_(teoria_dei_grafi))
- Robert W. Floyd: https://it.wikipedia.org/wiki/Robert_Floyd
- Algoritmo: <https://it.wikipedia.org/wiki/Algoritmo>



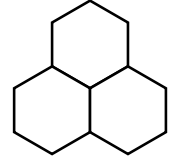


7. Piramide colorata

Sami mette insieme gli esagoni bianchi. Poi li dipinge con tre colori diversi.

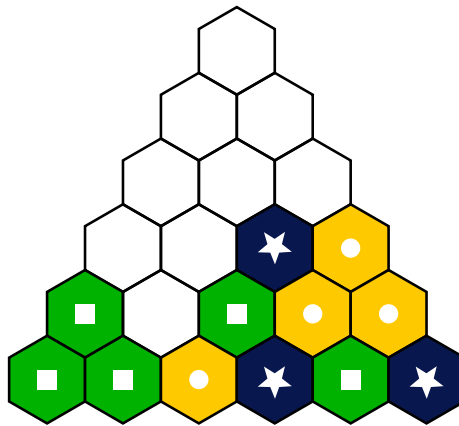
Sami vuole che quando tre esagoni si trovano esattamente insieme in questo modo (due in basso e uno in alto al centro), devono finire ...

- ... tutti e tre dello stesso colore o ...
- ... tutti e tre di colori diversi.



Sami ha messo insieme molti esagoni e ne ha già colorati alcuni.

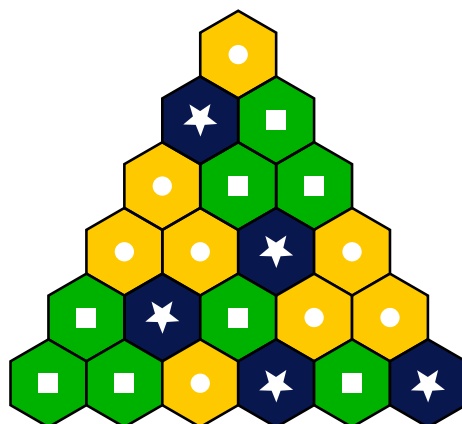
Colora tutti gli esagoni rimanenti come piace a Sami.





Soluzione

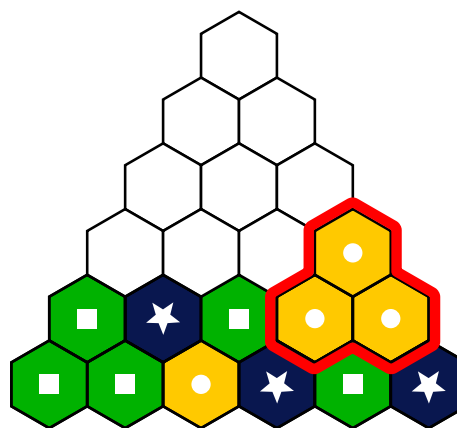
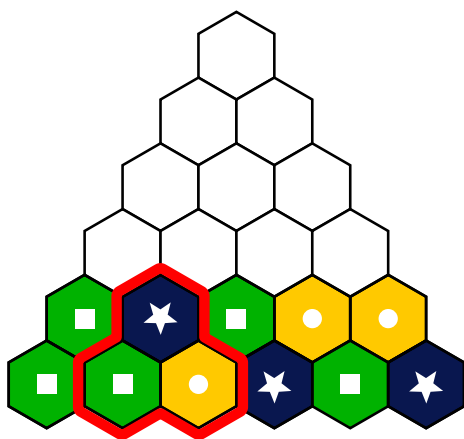
Questa è la soluzione giusta:



Non appena vengono colorati due esagoni vicini nella piramide di esagoni, il colore dell'esagono superiore viene fissato:

Se entrambi hanno colori diversi, l'esagono sopra riceve il terzo colore. Ad esempio, l'esagono bianco più basso è dipinto di blu

Se entrambi hanno lo stesso colore, anche l'esagono sopra di esso è dipinto di quel colore. Quindi anche l'esagono sopra i due gialli è dipinto di giallo.



In questo modo potrai colorare gli esagoni rimanenti in fila, dal basso verso l'alto, uno dopo l'altro, proprio come piace a Sami.

Questa è l'informatica!

Come si risolve questo compito? Quando colori un esagono, esegui un'azione. Per scegliere l'azione giusta (con il colore giusto), devi guardare gli esagoni sottostanti e verificare quale *condizione* soddisfano: se hanno lo stesso colore o colori diversi. Questo controllo, con le azioni successive, viene *ripetuto*, cioè per ogni esagono ancora bianco che si trova sopra due esagoni già colorati.



Azioni, condizioni, ripetizioni: questi sono gli elementi di base di qualsiasi *algoritmo*, cioè una procedura descritta con precisione che può essere realizzata come programma per un computer. Quindi, per risolvere questo compito, hai inventato un algoritmo. Questo è uno dei compiti più importanti degli informatici: inventare algoritmi o utilizzare algoritmi già inventati e convertirli in programmi per computer al fine di risolvere compiti e problemi di elaborazione automatica delle informazioni.

Parole chiave e siti web

- Algoritmo: <https://it.wikipedia.org/wiki/Algoritmo>
- Selezione: [https://it.wikipedia.org/wiki/Selezione_\(informatica\)](https://it.wikipedia.org/wiki/Selezione_(informatica))
- Ciclo: https://it.wikipedia.org/wiki/Struttura_di_controllo#Ciclo_for



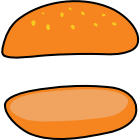








8. Ricetta hamburger

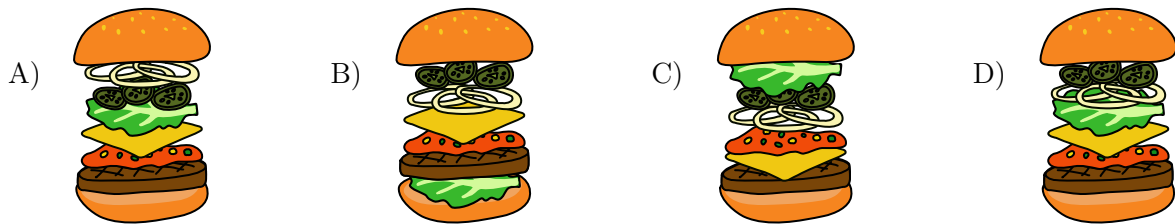
La castorina Jess prepara gli hamburger. Per farli segue tre regole:

1. la salsa è direttamente sulla carne.
2. la carne e il formaggio sono sotto i cetrioli, la lattuga e le cipolle.
3. le cipolle non toccano il panino.

Ingredienti dell'hamburger:

Panino	Carne	Salsa	Cetrioli	Lattuga	Cipolle	Formaggio
						

Quale hamburger è composto secondo le tre regole?





Soluzione



La risposta corretta è D.

Per trovare la soluzione, è necessario controllare ogni hamburger per vedere se è messo insieme in modo da seguire tutte e tre le regole.

- A) Questo hamburger segue le regole 1 e 2. Ma le cipolle toccano il panino, quindi non rispetta la regola 3.
- B) Questo hamburger segue la regola 1. Ma la lattuga è sotto la carne e il formaggio, quindi la regola 2 non è stata rispettata.
- C) Questo hamburger segue la regola 2 perché la carne e il formaggio sono sotto i cetrioli, la lattuga e le cipolle. Inoltre, questo hamburger di castoro segue la regola 3 perché le cipolle non toccano il panino. Tuttavia, la salsa non viene versata direttamente sulla carne. Pertanto, la regola 1 non è stata rispettata.
- D) Questo hamburger soddisfa tutte le regole.

Questa è l'informatica!

Gli hamburger in questo compito sono realizzati secondo tre regole. Per ogni hamburger che prepara, la castorina Jess deve seguire ognuna delle tre regole. Se non rispetta una sola delle regole, l'hamburger non è giusto. Ognuna delle tre regole è una condizione che deve essere soddisfatta affinché ogni hamburger sia giusto.

In informatica, il controllo dei vincoli è spesso usato per scoprire se una soluzione segue tutte le regole date. In questo controllo, si collegano tutte le regole (condizioni) con l'operatore *E*. Ciò significa che tutte le regole (condizioni) devono essere soddisfatte contemporaneamente.

Verificare se una determinata soluzione soddisfa tutti i vincoli è un compito fondamentalmente diverso dal trovare una possibile soluzione. Si tratta del cosiddetto *problema di soddisfacimento di vincoli*. Nella maggior parte dei casi, è molto più difficile trovare una soluzione che soddisfi tutti i vincoli che verificare se una soluzione soddisfa tutti i vincoli. Questo vale anche per un computer.

Parole chiave e siti web

- Programmazione a vincoli: https://it.wikipedia.org/wiki/Programmazione_a_vincoli
- Problema di soddisfacimento di vincoli:
https://it.wikipedia.org/wiki/Problema_di_soddisfacimento_di_vincoli
- Congiunzione logica: https://it.wikipedia.org/wiki/Congiunzione_logica



- NP: [https://it.wikipedia.org/wiki/NP_\(complessità\)](https://it.wikipedia.org/wiki/NP_(complessità))

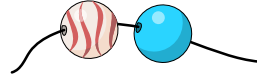




9. Collana da marinaio

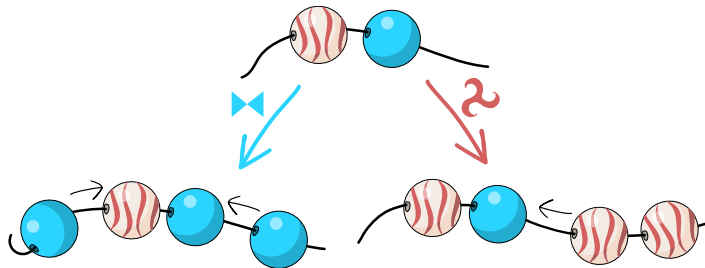
Ecco il manuale per la collana da marinaio di Monika con perline a onda bianche e rosse e perline blu semplici.

Inizia sempre con una perline a onda e una perline blu in questo ordine:



Poi puoi estendere la collana da marinaio,

- aggiungendo una perline blu a ciascuna estremità della stringa (↔)
- oppure aggiungendo due perline a onda all'estremità destra della stringa (↷)



Puoi eseguire queste azioni più volte per creare collane sempre più lunghe.

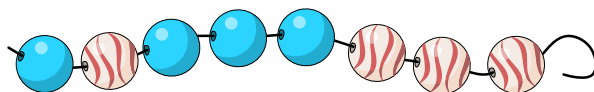
Quale delle seguenti collane **non** è una delle collane da marinaio di Monika?

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














Soluzione

D è la risposta corretta.



Puoi risolvere il problema in diversi modi.

Per esempio, trovando prima le due perline iniziali di ogni collana ed eseguendo poi una serie di azioni  e .

- Nella collana A, puoi iniziare con la seconda e la terza perline e poi eseguire le azioni  -  - .
- Per la collana B, puoi iniziare con la terza e la quarta perline e poi eseguire le azioni  -  - .
- Per la collana C, puoi iniziare con la seconda e la terza perline e poi eseguire le azioni  -  - .
- Tuttavia, se guardi la collana D, la seconda e la terza perline devono essere l'inizio. L'azione B può essere eseguita una volta, ma dopo di essa non ci sono altre azioni per ottenere il resto della catena.

Questo approccio non funziona bene se la collana è molto lunga e ha molte possibili perline di partenza. In questo caso, un approccio decostruttivo potrebbe funzionare meglio. In questo caso rimuovi ripetutamente le perline eseguendo l'azione B o l'azione W al contrario, finché non rimangono solo due perline.

Una terza strategia si avvale della *parità*. Secondo le istruzioni della collana del marinaio, c'è sempre un numero dispari di perline blu e un numero dispari di perline ondulate rosse e bianche («parità dispari»). Capisci perché?

La collana D ha un numero pari di entrambi i tipi di perline e quindi non può essere una delle collane da marinaio di Monika.

Questa è l'informatica!

In questa attività puoi infilare le perline solo alle estremità della collana. Non puoi inserire una perline al centro. Inoltre, non puoi rimuovere una perline dal centro senza aver prima sfilato le perline dall'estremità della collana.

Questo tipo di struttura di memoria, in cui è possibile aggiungere e rimuovere facilmente elementi alle estremità ma non al centro, è chiamata in informatica *coda a doppia estremità* o *coda deque* (deque si pronuncia come «deck»).

Le code deque possono essere utilizzate per memorizzare la cronologia del browser, per programmare i lavori di stampa e anche per verificare la validità delle espressioni matematiche. Ad esempio, il



controllo della corrispondenza delle parentesi può essere fatto più o meno nello stesso modo in cui si controlla se una collana è una delle collane di marinaio di Monika.

Parole chiave e siti web

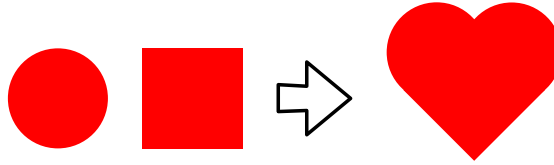
- double-ended queue: <https://it.wikipedia.org/wiki/Deque>





10. Cuore composto

Tina ha due forme: un cerchio e un quadrato. Li trasforma in un cuore.



Per farlo, utilizza queste tre trasformazioni:

- *gira*: gira una forma quanto si vuole.
- *sposta*: sposta una forma quanto si vuole.
- *raddoppia*: raddoppiare una forma in modo che entrambe rimangano nello stesso posto.

Cosa ha fatto e in che ordine?

- A) *raddoppia* il cerchio, *gira* il quadrato, *sposta* il cerchio, *sposta* il cerchio
- B) *raddoppia* quadrato, *gira* quadrato, *sposta* quadrato, *sposta* cerchio
- C) *raddoppia* cerchio, *gira* cerchio, *sposta* cerchio, *sposta* quadrato
- D) *sposta* cerchio, *sposta* cerchio, *raddoppia* cerchio, *sposta* quadrato


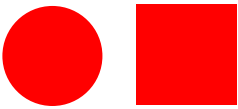
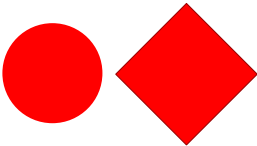
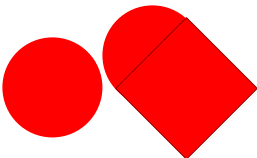
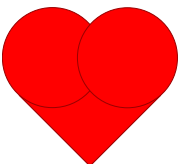


Soluzione

Se si osserva attentamente il cuore, si noter  che   composto da due cerchi e da un quadrato ruotato di 1/8. Quindi   necessario un «raddoppia cerchio» nelle trasformazioni, in modo da avere due cerchi, e un «gira quadrato», in modo da poter girare il quadrato di 1/8. Questo elimina le risposte B), C) e D), perch :

- Nella risposta B) viene raddoppiato un quadrato e non un cerchio.
- Nella risposta C) viene ruotato un cerchio, ma non il quadrato.
- Nella risposta D) nessuna forma viene ruotata.

Ma la risposta A)   corretta? Le forme devono ancora essere spostate! Le trasformazioni seguenti sono date:

- Questo: 
- diventa  raddoppiando il cerchio
- diventa  girando il quadrato
- diventa  spostando il cerchio
- diventa  spostando il cerchio

Pertanto, la risposta A) raddoppia il cerchio, gira il quadrato, sposta il cerchio, sposta il cerchio   corretta.

Questa   l'informatica!

Nei programmi di modifica delle immagini   possibile effettuare molte trasformazioni diverse con un'immagine. In questo compito, si tratta di trasformazioni come la rotazione, lo spostamento o il raddoppio. Ma questo da solo non basta: bisogna anche dire al computer, per esempio, di quanto ruotare una forma o dove spostarla.

Si potrebbe descrivere il modo in cui disegnare un cuore da un cerchio e da un quadrato in un testo pi  lungo. In informatica, tuttavia,   meglio utilizzare il minor numero possibile di trasformazioni di base, che poi si ripetono o si eseguono in modo diverso. Si parla di generalizzazione quando



si sviluppano soluzioni generali a partire da esempi specifici. Tali comandi potrebbero essere, ad esempio:

- Ruotare una forma: ruotare la forma, fino a che punto.
- Spostare una forma: spostare la forma, dove
- Raddoppiare una forma: doppia forma

Il programma di modifica delle immagini di Tina può sembrare insolito: invece di salvare l'immagine come *pixel* come nelle foto, viene salvata una descrizione della forma (ad esempio «cerchio, raggio 2 cm, colore di riempimento rosso»). In questo modo è possibile sovrapporre due forme, come i due cerchi, e spostare successivamente una di esse senza che quella inferiore venga sovrascritta. Questo tipo di grafica si chiama *grafica vettoriale*. Vengono spesso utilizzati quando si devono disegnare forme astratte di alta qualità. Gli altri elementi grafici utilizzano la *grafica a pixel* e spesso sono foto o disegni fotorealistici.


Parole chiave e siti web

- Pixel: <https://it.wikipedia.org/wiki/Pixel>
- Grafica raster: https://it.wikipedia.org/wiki/Grafica_raster
- Grafica vettoriale: https://it.wikipedia.org/wiki/Grafica_vettoriale





11. Bulloni e dadi

Ben è alla catena di montaggio e lavora i componenti: dadi  e bulloni .




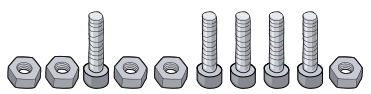
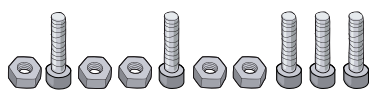
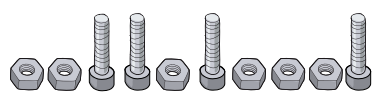
Ben segue rigorosamente la seguente procedura:

- Ben prende il componente successivo dalla catena di montaggio.
- Quando Ben ha preso un dado dalla catena di montaggio, lo mette nel secchio.
- Quando Ben ha preso un bullone dalla catena di montaggio, prende un dado dal secchio, lo avvita sul bullone e mette il pezzo finito nella scatola.

In questa procedura possono verificarsi due errori:

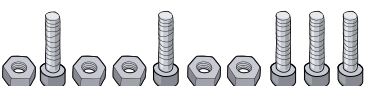
1. Ben prende un bullone dalla catena di montaggio, ma nel secchio non c'è nessun dado da avvitare.
2. Ben ha lavorato tutti i componenti della catena di montaggio, ma ci sono ancora dadi nel secchio.

Il secchio per i dadi è sufficientemente grande e vuoto all'inizio. Quale delle sequenze di dadi e bulloni può essere elaborata da Ben da sinistra a destra senza commettere errori?

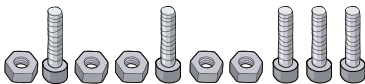

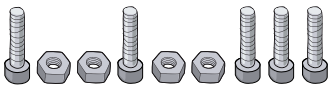




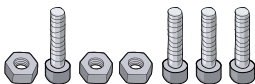


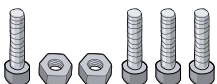


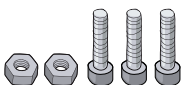


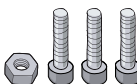



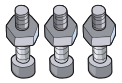


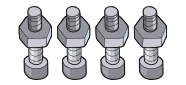
- A) 
- B) 
- C) 
- D) 




Soluzione

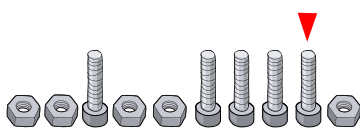
La risposta corretta è C) 

La tabella mostra lo stato della scatola per i pezzi finiti, del secchio per i dadi e della catena di montaggio.

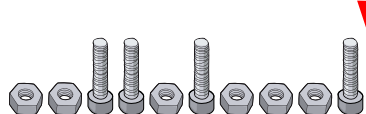
Scatola	Secchio	Catena di montaggio
<i>vuota</i>	<i>vuoto</i>	
<i>vuota</i>		
	<i>vuoto</i>	
		
		
		
		
		
		
	<i>vuoto</i>	<i>vuota</i>

Perché le altre risposte sono sbagliate?

A)  porta a un errore nella posizione contrassegnata. Poi Ben ha preso un bullone, ma nel secchio non c'è più il dado.

B)  porta a un errore nella posizione contrassegnata. Finora Ben ha avvitato 4 dadi su quattro bullone. Quindi il secchio è vuoto. Ma ora ha preso un quinto bullone per il quale non ha più un dado.



D)  porta a un errore dopo l'elaborazione dell'intera sequenza. Questo perché sono stati avvitati 4 dadi su 4 bulloni e sono rimasti 2 dadi.

Questa è l'informatica!

Ben elabora i componenti che vengono consegnati uno per uno dalla catena di montaggio. Nel processo, utilizza un grande secchio per conservare temporaneamente i dadi. Una disposizione simile viene utilizzata in *informatica teorica* come modello per gli *algoritmi* in grado di risolvere una certa classe di problemi: automi a pila.

Un automa a pila elabora i dati (numeri o caratteri) che riceve in ingresso uno per uno. Ha un'unica memoria infinita, una pila. A differenza del secchio nel compito, gli elementi della pila hanno un certo ordine e si può togliere da una cantina solo l'elemento che si è messo per ultimo («last in first out», LIFO). Un automa di pila può essere utilizzato per riconoscere un *linguaggio libero dal contesto*.

In informatica, un linguaggio è un insieme di stringhe formate secondo determinate regole. Un tipo semplice di linguaggio è il linguaggio libero dal contesto. Un esempio di linguaggio libero dal contesto è costituito da tutte le espressioni ben formate di parentesi. In un'espressione ben formata, ogni parentesi aperta viene chiusa. Le espressioni ben formate sono, ad esempio, $((()))$ e $(() ())$. Non ben formati, invece, sono $(((())$ e $() (()$. Si può pensare ai dadi e ai bulloni del compito come a delle parentesi di apertura e chiusura. Quindi Ben elabora una sequenza di componenti sulla catena di montaggio senza errori solo se rappresenta un'espressione di parentesi ben formata. La verifica delle espressioni di parentesi è un compito importante di un compilatore che traduce i testi dei programmi in programmi eseguibili. Questo perché le chiamate di funzione annidate e le espressioni aritmetiche con parentesi sono presenti nei testi dei programmi della maggior parte dei linguaggi di programmazione.

Parole chiave e siti web

- Informatica teorica: https://it.wikipedia.org/wiki/Informatica_teorica
- Automa a pila: https://it.wikipedia.org/wiki/Automa_a_pila
- Linguaggio libero dal contesto:
https://it.wikipedia.org/wiki/Linguaggio_libero_dal_contesto



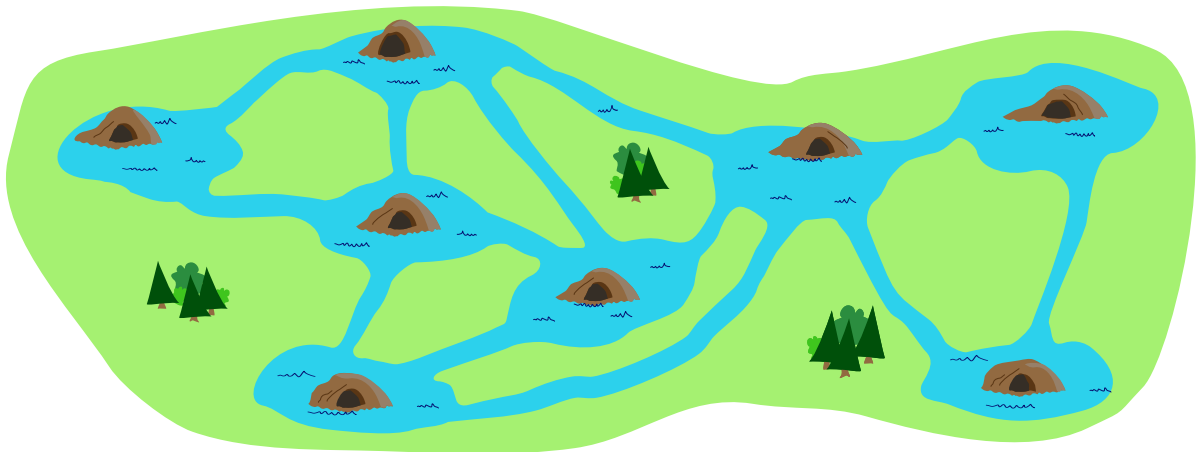


12. I vicini di Lili

Sulla mappa si possono vedere i castelli di otto castori. Due castori sono vicini di casa se un canale collega i loro castelli.

- Lili, Simon e Peter hanno ciascuno quattro vicini.
- Simon e Peter sono gli unici vicini di Nina.

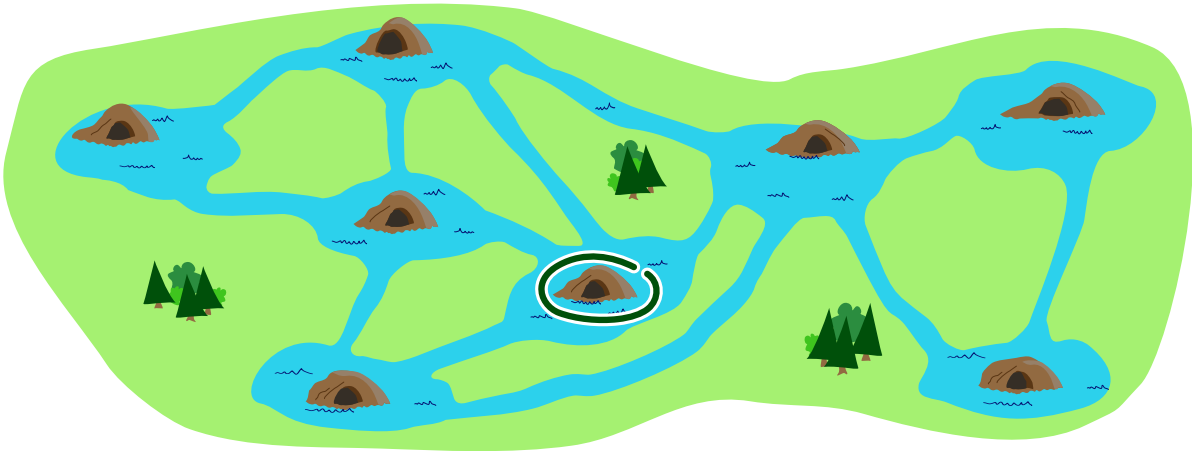
In quale castello vive Lili?



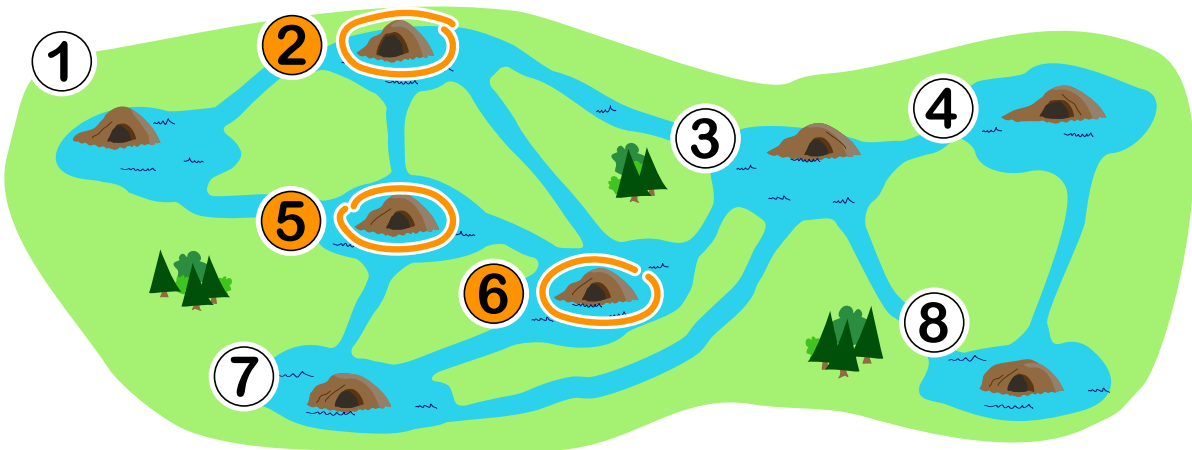


Soluzione

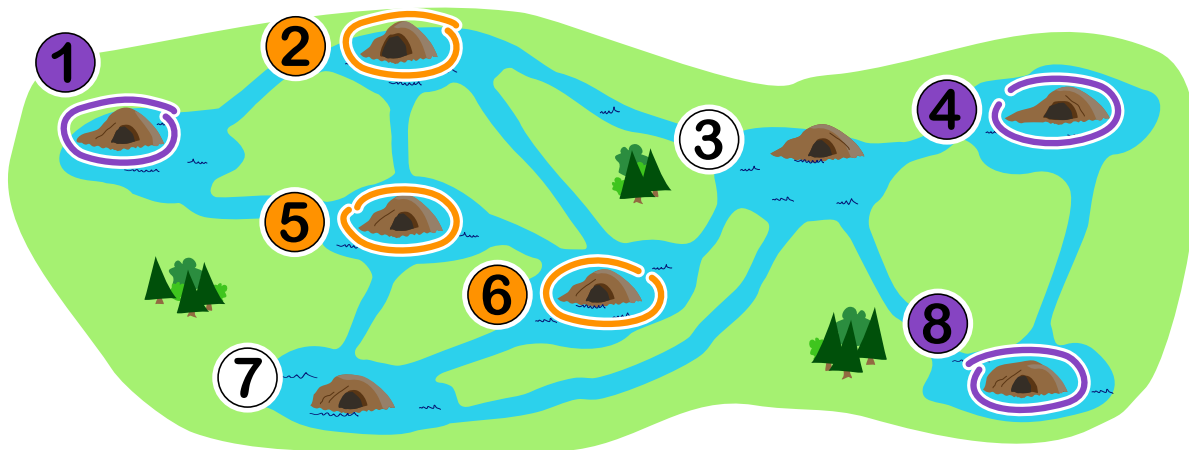
La soluzione corretta è:



Per risolvere il problema, è necessario concentrarsi sui canali tra i castelli. Dobbiamo identificare i castelli in cui vivono Lili, Peter o Simon. Poiché tutti hanno 4 vicini, ci devono essere esattamente quattro canali che partono da ciascuno dei loro castelli. Ci sono tre castelli di questo tipo: 2, 5 e 6.



Di conseguenza, Lili, Peter e Simon vivono ciascuno in uno di questi tre castelli. Ora dobbiamo scoprire in quale dei tre castelli vive Lili. Le altre due informazioni si riferiscono al castello di Nina. Da questi possiamo concludere che dal suo castello partono esattamente due canali. Quindi Nina vive in uno di questi castelli: 1, 4 o 8.



Siccome sappiamo che Simone e Pietro sono i due vicini di casa di Nina, possiamo concludere che

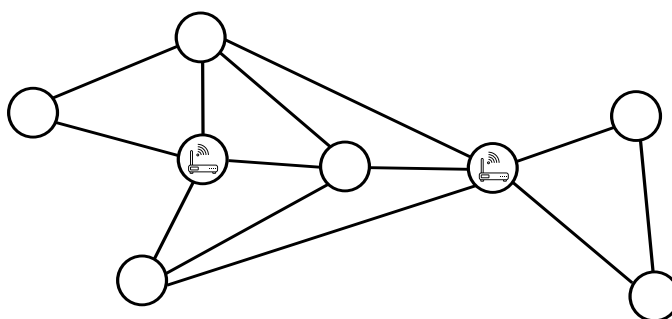
- Nina vive nel castello 1
- Simon e Peter vivono nei castelli 5 e 7 (o viceversa).

Quindi c'è solo un castello da cui partono quattro canali, che può essere il castello di Lili. È il castello 6!

Questa è l'informatica!

In questo compito, due castelli sono collegati da un canale. L'insieme dei castelli e dei canali forma una rete che mostra le relazioni tra tutti i castelli. Una tale rete di relazioni tra oggetti è chiamata *grafo* in informatica e matematica. Un grafo può essere considerato come un *insieme* di *vertici* collegati da *archi*. In questo compito, i castelli rappresentano i vertici e i canali gli archi.

Lo studio dei grafi è chiamato *teoria dei grafi*. Può essere utilizzato per modellare le relazioni a coppie tra gli oggetti. I grafi sono modelli matematici di strutture simili a reti in natura e in tecnologia. Ne sono un esempio le strutture sociali, le reti stradali, le reti informatiche, i circuiti elettrici, le reti di distribuzione o le molecole chimiche. I grafi possono essere utili per descrivere e risolvere i *problemi di rete*, come ad esempio trovare un buon posto per un router in un edificio o assicurarsi che ogni stanza di una casa abbia un segnale Wi-Fi forte.





Parole chiave e siti web

- Grafo: <https://it.wikipedia.org/wiki/Grafo>
- Insieme: <https://it.wikipedia.org/wiki/Insieme>
- Vertice: [https://it.wikipedia.org/wiki/Vertice_\(teoria_dei_grafi\)](https://it.wikipedia.org/wiki/Vertice_(teoria_dei_grafi))
- Arco: https://it.wikipedia.org/wiki/Glossario_di_teorica_dei_grafi#Arco



A. Autori dei quesiti

 Esraa Almajhad


 Leo Barichello

 Liam Baumann

 Wilfried Baumann

 Linda Björk Bergsveinsdóttir


 Tobias Berner

 Marta J. Burzanska

 Sarah Chan

 Byeonggyu Cho

 Kris Coolsaet

 Darija Dasović

 Christian Datzko

 Susanne Datzko


 Justina Dauksaite

 Nora A. Escherle

 Gerald Futschek

 Christian Giang

 Mark Edward M. Gonzales

 Yasemin Gülbahar


 Benjamin Hirsch

 Alisher Ikramov

 Thomas Ioannou

 Dauksaite Justina

 Hakin Kim

 Jihye Kim

 Seulki Kim


 Vaidotas Kinčius

 Regula Lacher

 Taina Lehtimäki

 Marielle Léonard

 Karolína Miková

 Jelena Milojkovic


 Madhavan Mukund

 Ágnes Erdősne Németh

 Zsuzsa Pluhár

 Wolfgang Pohl

 John-Paul Pretti


 Le Quang Quan

 Chris Roffey


 Kirsten Schlüter

 Bernadette Spieler

 Goran Sukovic


 Monika Tomcsányiová

 Ahto Truu

 Troy Vasiga

 Florentina Voboril

 Michael Weigend

 Kyra Willekes



B. Sponsoring: concorso 2022

HASLERSTIFTUNG

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



Kanton Zürich
Volkswirtschaftsdirektion
Amt für Wirtschaft und Arbeit

Standortförderung beim Amt für Wirtschaft und Arbeit Kanton Zürich



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



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

Musée des transports, Lucerne



i-factory (Musée des transports, Lucerne)



<http://senarclens.com/>

Senarclens Leu & Partner



<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

Scuola universitaria professionale della Svizzera italiana

<http://www.supsi.ch/home/supsi.html>

La Scuola universitaria professionale della Svizzera italiana (SUPSI)

SUPSI



C. Ulteriori offerte



La Fiamma IT: <https://it-feuer.ch/it/>

In Svizzera, numerose organizzazioni si impegnano per la formazione delle giovani leve nell'ambito dell'informatica. L'iniziativa «La Fiamma IT» vuole unire queste forze e contribuire insieme a diffondere il tema nell'opinione pubblica in tutta la Svizzera. La fiamma IT presenta numerose offerte rivolte sia ai docenti che agli studenti.



CoetryLab: <https://www.coetry-lab.org/>

Il team del CoetryLab (Zürich) vuole dare ai bambini e ai giovani l'accesso alla programmazione e ai media. Il Coetry-Lab vuole essere il luogo di sperimentazione e progettazione extrascolastica e aprire il mondo del coding a tutti. Le loro idee possono essere realizzate in modo creativo e siti web, applicazioni, giochi e molto altro possono essere sviluppati in team o da soli.



Roteco: <https://www.roteco.ch/it/>

Il progetto Roteco consiste in una comunità di insegnanti desiderosi di preparare gli allievi per la società digitale. In questa comunità gli insegnanti trovano, sviluppano e si scambiano attività didattiche inerenti la robotica educativa e più in generale le scienze informatiche pronte da essere utilizzate in classe e vengono informati con le ultime novità e corsi in questi campi.

010100110101011001001001
 010000010010110101010011
 010100110100100101000101
 001011010101001101010011
 010010010100100100100001

SS!

www.svia-ssie-ssii.ch
 schweizerischerverein für informatikind
 erausbildung//société suisse pour l'infor
 matique dans l'enseignement//società sviz
 zera per l'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.