Indice

Le biglie - 3 punti

Quattro amici hanno un grosso sacchetto di biglie e vogliono provare a distribuirsele in modo che ogni gruppetto di loro ne abbia un numero diverso dagli altri. Scoprono subito che distribuire 10 biglie (per esempio: $1+2+3+4$) non va bene, perché $1+2=3$, invece le somme devono essere tutte diverse. Qual è il minimo numero totale di biglie da distribuire? **Spiegate la vostra risposta!**

Soluzione La risposta è **15**. Il testo fa riferimento a "un grosso sacchetto di biglie": è inteso che i quattro amici ne abbiano a disposizione un numero sufficiente, almeno tante quante ne prevede la risposta, e non è detto che debbano distribuirsele tutte! Le richieste da rispettare sono due:

1. che ogni gruppetto di amici abbia un numero di biglie diverso da quello di ciascun altro possibile gruppetto;

2. che il numero di biglie distribuite sia il minimo numero che rispetti la richiesta 1.

Un modo per ottenere somme tutte diverse è sicuramente quello di distribuire le biglie dandone un certo numero al primo e poi via via un numero maggiore della somma di quelle già distribuite.

Chiamiamo A, B, C e D i quattro amici. Se diamo una biglia ad A, due ($2>1$) a B, quattro ($4>1+2$) a C e otto ($8>1+2+4$) a D, per un totale di 15 biglie, soddisfiamo la richiesta 1.

Con questa distribuzione (1, 2, 4 e 8, cioè le prime quattro potenze di 2), si ottengono come somme nei diversi gruppi *tutti* i numeri interi da 1 a 15: 1 (A), 2 (B), 3 (A+B), 4 (C), 5 (A+C), 6 (B+C), ... , 15 (A+B+C+D).

Siamo cosí sicuri che sia rispettata anche la richiesta 2. Infatti non si possono ottenere 15 somme diverse con meno di 15 biglie: se il minimo fosse minore di 15 alcune somme dovrebbero essere uguali.

Perché i diversi gruppetti possibili dei 4 amici sono 15? Perché sono tanti quanti i sottoinsiemi *non vuoti* di un insieme di 4 elementi. I sottoinsiemi di un insieme di n elementi sono 2^n (considerando anche l'insieme vuoto), ossia il prodotto di n volte 2: nel nostro caso $2^4 = 2\times 2\times 2\times 2 = 16$. Se diamo 0 biglie al gruppetto di ...0 amici otteniamo i 16 valori da 0 a 15.

Per convincersi di questo fatto, basta immaginare che ad ognuno dei 4 (ovvero n) amici sia associato un *bit*, posto a valore 1 se egli fa parte del gruppetto considerato, a valore 0 altrimenti ... e le configurazioni possibili dei 4 (ovvero n) *bit* sono proprio 16 (ovvero 2^n).

Attenzione! In questo modo minimizziamo il *totale* delle biglie distribuite, non il *massimo* numero di biglie dato a un amico: distribuendo 3, 5, 6 e 7 biglie si ottengono ancora somme tutte diverse, ma il totale è 21.

I gettoni e i piattini - 4 punti

Su un tavolo ci sono 4 gettoni colorati sovrapposti: sopra un gettone rosso, poi uno blu, uno giallo e uno verde. Ci sono anche due piattini: dovete spostare i gettoni sui piattini mettendoli sempre uno sopra l'altro e spostando solo il gettone superiore. Non si possono rimettere i gettoni sul tavolo e si deve formare su un piattino la pila in ordine ... alfabetico: blu sopra, poi giallo, rosso, verde, *spostando il minimo numero di gettoni*. Descrivete la vostra soluzione con la sequenza delle mosse da effettuare: denotati con 1 e 2 i due piattini e con B, G, R, V i quattro colori, R1 significa: sposta il gettone rosso sul piattino 1, V2 sposta il gettone verde sul piattino 2, e cosí via. Qual è la vostra sequenza?

Soluzione Osserviamo che una volta che i gettoni sono stati tutti trasferiti dal tavolo ai piattini si può solo invertire il loro ordine spostandoli da un piattino all'altro, non potendo rimettere i gettoni sul tavolo. Occorre quindi ottenere su uno dei due piattini la pila in ordine alfabetico diretto o inverso, e poi eventualmente capovolgerla.

Ci sono ovviamente **due soluzioni simmetriche**, in cui i piattini 1 e 2 sono scambiati.

Inizialmente i gettoni sono disposti a formare una pila, con in alto il gettone rosso e in basso il gettone verde. Siamo quindi costretti a spostare i tre gettoni piú in alto per liberare il gettone verde; dobbiamo però fare attenzione a spostare questi tre gettoni in modo da metterli nell'ordine: in alto il rosso, poi il giallo e in basso il blu (ovvero nell'ordine inverso a quello desiderato). Cominciamo quindi col mettere il gettone rosso sul piattino 1 (R1) e il blu sul 2 (B2), poi il giallo deve andare sul blu (G2) e finalmente il rosso sul giallo (R2); ora spostiamo il verde sul piattino 1 (V1), rovesciamo sul piattino 1 la pila precedentemente formata sul piattino 2 e abbiamo fatto in tutto 8 mosse! Dunque

R1 B2 G2 R2 V1 R1 G1 B1,

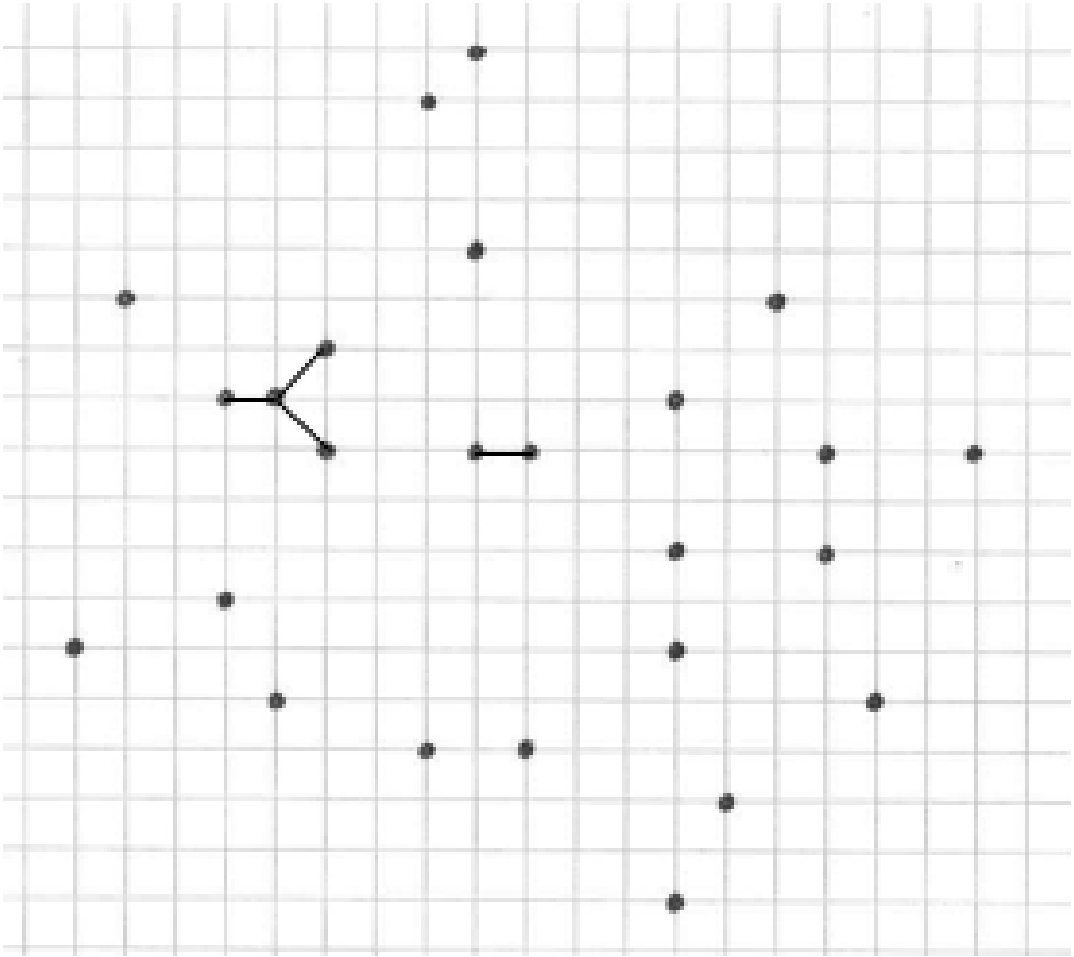
oppure, scambiando 1 con 2,

R2 B1 G1 R1 V2 R2 G2 B2.

Le semplici **pile**, sorprendentemente, sono strutture di dati fondamentali in informatica. Pensiamo, ad esempio, a un contabile che deve completare una certa pratica A. Quando ha bisogno di un'informazione che si può ottenere da un'altra pratica B, lascia in sospeso A sul tavolo e sopra di essa deposita la pratica B da consultare. Mentre esamina questa, può capitare che abbia bisogno di un dato che si trova nella pratica C, cosí lascia in sospeso anche B, prende la pratica C e la mette sopra le altre... Quando ha ottenuto quel che voleva, ripone la pratica C e riprende B (subito sotto nella pila) dal punto in cui l'aveva lasciata in sospeso, e cosí via finché la prima pratica, A, non è terminata. Proprio in questo modo opera di solito un computer quando esegue un programma: durante l'esecuzione può accadere che sia chiamato un sottoprogramma dalla cui esecuzione si vuole ottenere un certo risultato; ma questo sottoprogramma, a sua volta, può chiamarne un altro... Cosí, quando un processo (quello che sta in cima alla pila) termina e "sparisce", il suo risultato è trasmesso al "chiamante" (che sta subito sotto nella pila), il quale può riprendere dal punto in cui si era interrotto; e quando la pila sarà di nuovo vuota, l'esecuzione dell'intero programma sarà conclusa!

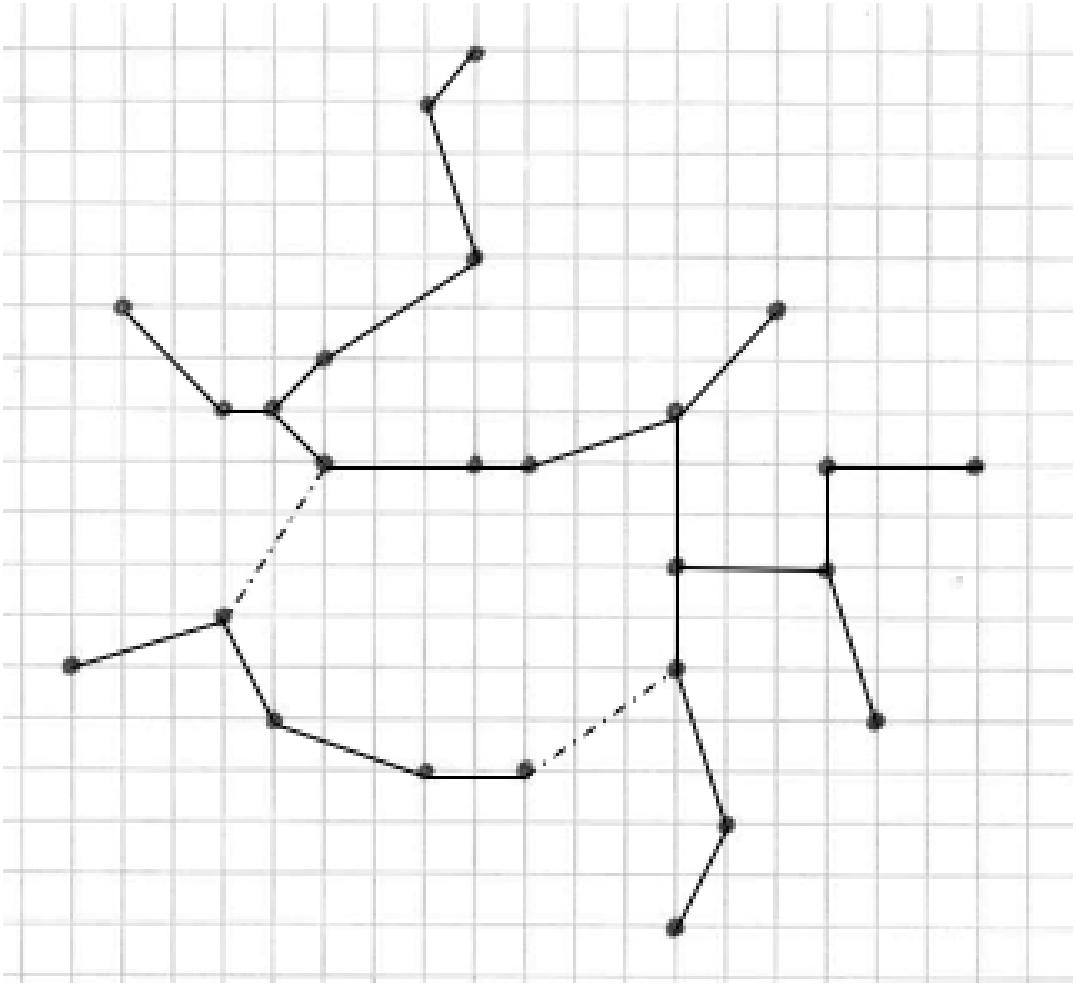
Gli elettricisti - 4 punti

Due elettricisti, Crusca e Primo, devono collegare con cavi elettrici 25 *punti luce* disposti come illustrato nella figura qui a lato, in modo che ciascun punto sia collegato, ma senza che esistano anelli e impiegando la minor lunghezza possibile di cavo elettrico. L'apprendista Primo pensa che sia un compito difficilissimo. L'esperto Crusca, però, gli spiega che basta collegare via via la coppia di punti piú vicini, come si è già cominciato a fare nella figura, evitando solo di formare anelli. Completate i collegamenti seguendo il consiglio di Crusca.



Soluzione L'obiettivo consiste nel collegare tutti i punti luce in modo che nessuna porzione di rete resti isolata dal resto, senza creare anelli: in altre parole, considerati due punti luce qualsiasi, il collegamento tra loro, costituito da uno o piú segmenti di filo, deve essere unico! Inoltre, la somma delle lunghezze di tutti i segmenti di filo utilizzati deve essere la minore possibile. Se in ogni punto luce immaginiamo sia collocata una lampadina, collegando allora

due punti luce qualsiasi ai morsetti di una batteria si illuminerà quell'unico percorso che li unisce, che – si osservi – può essere costituito da piú segmenti non allineati (e quindi non è in generale il piú breve costruibile tra i due punti considerati), proprio perché ciò che si deve minimizzare è la lunghezza totale dei collegamenti.

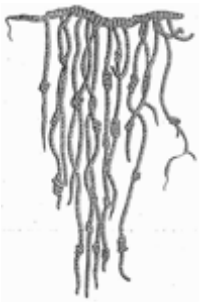


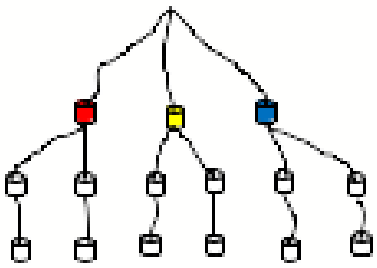
Nota. La soluzione non è unica. I due collegamenti tratteggiati chiudono un *anello*: bisogna scegliere *uno solo* dei due per completare la rete elettrica. La rete elettrica realizzata collega tutti i punti mediante una struttura ad albero. Una possibile difficoltà consisteva nel valutare le lunghezze dei collegamenti: se le misuriamo usando come unità il lato di ciascun quadretto, i due collegamenti tratteggiati, per esempio, avranno ciascuno lunghezza pari alla radice quadrata di $13 = 2^2 + 3^2$, per il *teorema di Pitagora*, applicato a un triangolo rettangolo di cateti 2 e 3 quadretti. Ma dovendo solo valutare se un lato è piú lungo di un altro non c'è alcun bisogno di calcolare radici: si possono semplicemente confrontare i quadrati. Cosí i nostri collegamenti tratteggiati hanno sicuramente lunghezza minore di 4, perché $4^2 = 16 > 13$. Non era comunque vietato usare un righello...

Il consiglio dell'esperto Crusca corrisponde in realtà al ben noto *algoritmo di Kruskal*, mentre un algoritmo alternativo è noto come *algoritmo di Prim*, il che dovrebbe giustificare i nomi degli elettricisti. Ma come procede l'algoritmo di Prim? Si parte da un punto arbitrario e lo si collega con un segmento al punto piú vicino, poi ad ogni passo successivo si collega un nuovo punto (finché ce ne sono): il piú vicino a qualcuno di quelli già collegati... è collegato con un segmento proprio a questo "qualcuno"! Cosí, alla fine, il risultato è ancora un albero (non orientato) che ha come nodi tutti i punti collegati. Ciò equivale a dire che la rete è connessa minimale (togliendo uno qualsiasi dei segmenti, la rete non è piú connessa, ossia "tutta d'un pezzo"), ed equivale pure a dire che la rete è aciclica massimale (aggiungendo un nuovo segmento tra due punti si crea inevitabilmente un anello, ossia un ciclo). Provate ad applicare l'idea di Prim al nostro problema, a partire da un punto a vostra scelta, e vi accorgerete che si ottiene ancora l'una o l'altra delle soluzioni di Kruskal: la somma delle lunghezze dei segmenti è sempre la minima!

Il quipu - 5 punti

Il ***quipu*** (si veda la figura a sinistra) è un groviglio di cordicelle annodate, usato nell'impero Inca per registrare numeri (la parola in lingua *quechua* significa *nodo*). Ispirandoci liberamente al quipu, supponiamo di avere tante cordicelle annodate in modo da formare un *albero* come quello della figura a destra, e di voler mettere in ogni nodo una perlina gialla, rossa o blu in modo tale che non ci siano colori uguali lungo nessuna cordicella e le cordicelle presentino *tutte le possibili sequenze* dei tre colori.

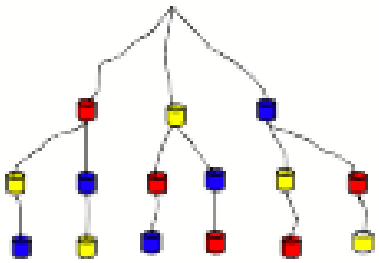




Primo quesito (2 punti) Tre delle 15 perline necessarie sono già colorate: colorate le altre (o indicate i colori con R, G e B) rispettando il vincolo sui colori.

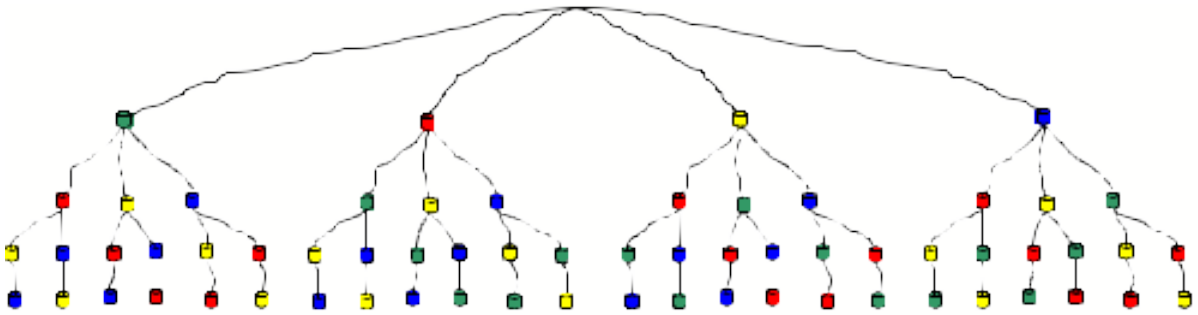
Secondo quesito (3 punti) Per un quipu con 4 colori diversi servono 64 perline. Qual è il numero di perline necessarie per un quipu con 5 colori diversi? **Giustificate la vostra risposta**.

Soluzione La soluzione al primo quesito era davvero facile e ne riportiamo a fianco una tra le molte possibili.



Per il quipu con 5 colori servono **325** perline. Ne occorrono 5 al primo livello, $5 \times 4 = 20$ al secondo, $20 \times 3 = 60$ al terzo e $60 \times 2 = 120$ al quarto e quinto livello. In totale

$5 + 20 + 60 + 240 = 325$. Oppure, denotato con $p(n)$ il numero di perline per il quipu con n colori, la *relazione di ricorrenza* per $p(n)$ è data da $p(1) = 1$ e $p(n) = n \times [p(n-1)+1]$, perché per formare il quipu con n colori si possono prendere n quipu con $n-1$ colori, aggiungendo poi una perlina alla *radice* degli n quipu. Dunque $p(4) = 4 \times [p(3) + 1] = 4 \times 16 = 64$ e $p(5) = 5 \times [p(4) + 1] = 5 \times 65 = 325$. L'indicazione che "per un quipu con 4 colori diversi servono 64 perline" (si veda la figura qui sotto) voleva esser un suggerimento su come vanno costruiti i nostri "quipu", che sono in realtà degli *alberi* (senza la perlina nel nodo radice). Le strutture ad albero sono assolutamente fondamentali in informatica: anche l'archiviazione dei file nei PC ha una struttura ad albero.



La macchina dell'abc - 6 punti

Il professor McCulloch ha costruito una strana macchina M, che è in grado di leggere delle parole e, a seconda della parola che ha letto, di scrivere eventualmente una nuova parola. La macchina *conosce solo l'abc*, ossia legge solo parole (senza senso) che contengano le lettere a, b o c. Se la parola comincia per a o è costituita da una sola lettera, la macchina non scrive niente. Se invece la parola comincia per b o c seguite da altre lettere allora la macchina scrive una parola secondo le regole descritte piú sotto.

Possiamo indicare con x una parola qualsiasi formata con le lettere a, b, c, e con bx una parola formata da b seguita dalla parola x; in modo analogo possiamo indicare con cx una parola che inizia per c. Indichiamo con $M(x)$ la parola che la macchina scrive quando legge la parola x. Ora possiamo descrivere il funzionamento della macchina con queste semplici regole:

$M(\mathrm{b}x) = x$
cioè la macchina scrive la parola che ha letto dopo la prima b;

$M(\mathrm{c}x) = M(x)\mathrm{b}M(x)$
cioè, se la parola comincia per c, allora la macchina legge la parola x che resta togliendo la c e scrive la parola $M(x)$, poi scrive b e poi di nuovo $M(x)$.

Ad esempio: $M(\mathrm{bac}) = \mathrm{ac}$, e $M(\mathrm{cba}) = \mathrm{aba}$ (poiché $M(\mathrm{ba}) = \mathrm{a}$).

Primo quesito (2 punti) Quale parola y produce sé stessa, cioè è tale che $M(y) = y$?

Secondo quesito (4 punti) Quale parola z produce $M(z) = z\mathrm{b}z$?

Soluzione del primo quesito La risposta è **cbc**. La parola y non potrà iniziare per a, dato che la macchina non scriverebbe nulla, né per b, dato che non scriverebbe la b. Possiamo far scrivere a M qualunque parola p: basta che legga bp. Dunque se $x = \mathrm{b}p$ avremo $M(x) = p$ e quindi $M(\mathrm{cb}p) = p\mathrm{b}p$, che può essere uguale a cb$p$ solo se $p = \mathrm{c}$. Ma in questo caso era forse piú facile trovare la soluzione per tentativi.

Soluzione del secondo quesito La risposta è **ccbcc**. Seguendo il ragionamento visto sopra si arriva a chiedersi se possa essere cb$p$ = cb$p$ b cb$p$: ma questo è impossibile! Sí, ma solo per parole che iniziano per cb! Se però una parola inizia per cc, la seconda regola andrà applicata *due volte*, e quindi

$$M(\mathrm{ccb}p) = M(\mathrm{cb}p)\ \mathrm{b}\ M(\mathrm{cb}p) = p\mathrm{b}p\ \mathrm{b}\ p\mathrm{b}p,$$

che è proprio della forma z b z con $z = p\mathrm{b}p = \mathrm{ccb}p$, uguaglianza possibile se $p = \mathrm{cc}$ e $z = \mathrm{ccbcc}$.

Osservazione Questi quesiti, ispirati da analoghi nel libro di Raymond Smullyan *Donna o tigre?* (recentemente ristampato nella collana per le edicole *Sfide matematiche* n. 27, RBA Italia, 2009), sono indubbiamente piuttosto insoliti e difficili, anche se abbiamo cercato di semplificarli (forse troppo). Qualcuno si chiederà giustamente che cosa stampi la macchina leggendo per esempio ca. Dovrebbe stampare M(a) b M(a), ma leggendo a la macchina non stampa nulla: allora stamperà b o non stamperà nulla? Bisognerebbe precisarlo oppure, come abbiamo tacitamente supposto, limitarci a considerare quelle parole x per cui la macchina stampa qualcosa, e precisamente $M(x)$.

Cerca la parola mancante - al massimo 20 punti

Questa prova va svolta su carta. Se volete, potete usare Internet per fare delle ricerche: in questo caso riportate le fonti che avete usato!

Nei testi seguenti alcune parole non sono state stampate. Otterrete un punto per ogni parola corretta che inserite negli spazi.

La ricerca per costruire macchine che imitino il comportamento umano ha una lunga storia, ma molti concordano sul fatto che l'origine dell' _____ *risale al 1950, quando venne pubblicato "Computing machinery and intelligence" di* _____. *In questo lavoro l'autore sostenne la tesi che le macchine potevano essere* _____ *in modo da mostrare un comportamento intelligente. Nell'articolo venne anche presentato un* _____, *divenuto poi famoso col nome del suo inventore, per determinare se una macchina sia in grado di pensare.*

Nel linguaggio naturale gli elementi hanno spesso nomi formati da piú parole, come in "tariffa oraria ridotta". Tuttavia, quando si esprimono gli _____ *in un linguaggio di programmazione oppure in pseudocodice, i nomi composti possono complicare la descrizione. L'esperienza ha insegnato che è meglio identificare ogni elemento con una singola* _____ *di testo. Un metodo molto diffuso per farlo usa la* _____ *per collegare le parole creando nomi come "tariffa_oraria_ridotta". Un altro metodo consiste nell'utilizzare le lettere* _____ *per aiutare il lettore a comprendere nomi complessi formati da piú parole, come in "tariffaOrariaRidotta". Questa tecnica è chiamata* _____ *perché ricorda le gobbe di un* _____. *Gli appassionati di*

computer che desiderano sperimentare con i componenti interni di un _____ *dovrebbero provare* _____, *progettato originariamente da Linus Torvald quando era studente all'università di Helsinki. Si tratta di software* _____ *e quindi disponibile gratuitamente insieme con il suo* _____ *sorgente e con la documentazione. Questo ha favorito la sua diffusione e la sua affidabilità e l'ha reso famoso come alternativa agli altri sistemi* _____ *in commercio, come Microsoft Windows o Mac OS.*

Ormai si è definitivamente imposta l'usanza di chiamare _____ *la posta elettronica indesiderata, di contenuto commerciale o truffaldino. Si tratta di un fenomeno estremamente fastidioso, anche se i* _____ *(programmi speciali che riconoscono la "spazzatura" e la segnalano o addirittura rimuovono dalla cartella della posta in arrivo) possono alleviare il problema. Probabilmente il termine, che in origine indica un tipo di* _____ *in scatola molto popolare in Inghilterra, si è diffuso grazie ad un vecchio sketch del gruppo comico* _____ *ambientato in un locale nel quale ogni pietanza proposta dalla cameriera era proprio a base di* _____.

Soluzioni.

intelligenza artificiale
Alan Turing
programmate
test

algoritmi
stringa (o parola)
sottolineatura
MAIUSCOLE
CamelCase ("carattere a cammello")
cammello

sistema operativo
Linux
libero
codice
proprietari (o operativi)

spam
filtri
carne
Monty Python
spam

Il Grande Torneo Transitivo - al massimo 15 punti

Questa prova va svolta usando il computer. La soluzione deve però essere riportata nell'apposito spazio in fondo alla pagina.

Alcuni tra gli ultimi vincitori del prestigioso "Premio Turing" (l'equivalente del Premio Nobel per le discipline informatiche) hanno deciso di sfidarsi in un torneo di ping-pong per decidere una volta per tutte una graduatoria condivisa dei loro contributi.

I nostri eroi, ormai un po' anziani, non vogliono giocare troppe partite, ma hanno calcolato che, seguendo un'opportuna strategia, anche nel *caso pessimo* 7 incontri sono sufficienti per concludere il torneo senza ex-aequo, grazie alla *proprietà transitiva*: se a batte b e b batte c, allora a e b precederanno c in classifica.

La *legge di Murphy* ("Se qualcosa può andar storto, lo farà!") sembra però dominare l'esito degli incontri! Riuscite a ottenere una classifica univoca con al massimo 7 incontri?

Per iniziare la prova, cliccate sul bottone "Torneo". Cliccate sulle immagini per selezionare i giocatori di una partita, che viene giocata premendo il pulsante "Disputa la partita". Quando ritenete che la graduatoria sia univoca premete "Valuta la classifica".

Riportate nello spazio seguente l'elenco delle partite disputate (con relativo vincitore) e la classifica finale ottenuta.

Soluzione Qual è la *strategia* da seguire per disputare il minor numero di partite? Il primo incontro può essere scelto arbitrariamente, cosí come il secondo, pur di scegliere tra i giocatori che non hanno già giocato. Se è successo che A abbia battuto B e B abbia battuto C, allora non avrà luogo alcun incontro tra A e C: nella classifica finale, A precederà B e B precederà C, e quindi A precederà C.

Adesso abbiamo due vincitori (chiamiamoli A e B) e due sconfitti (C e D). Scegliamo per esempio gli sconfitti e facciamoli giocare tra loro: se A aveva battuto C e ora C batte D avremo una classifica parziale ACD e inoltre sappiamo che B dovrà precedere D (avendolo battuto).

Il quinto giocatore E non ha ancora giocato e abbiamo già disputato 3 incontri: con chi conviene farlo giocare adesso? Col giocatore C a *metà classifica* (parziale), in modo che se E vince incontrerà poi A, mentre se perde incontrerà D, *ma non entrambi*. In questo modo E disputa due incontri e il totale sale a 5.

Ora la classifica parziale comprende 4 giocatori e supponiamo sia $ACDE$: siamo fortunati, per collocare B che precede comunque D (avendolo battuto all'inizio) dovremo al piú disputare i due incontri di B con A e C, e il totale è 7.

Se la classifica parziale fosse invece $ACED$ basterebbe disputare l'incontro di B con C, e poi se B vince gioca con A, altrimenti con E, ma non con entrambi, e ancora bastano 7 incontri. La stessa *strategia della "metà classifica"* funziona naturalmente anche se in precedenza E batte C, e può essere estesa a tornei con piú giocatori (ma le cose diventano un po' piú complicate e non è garantito in generale di ottenere sempre il minimo numero di incontri necessari!).

È un po' piú complicato anche fare in modo che si applichi la "legge di Murphy", ossia rendere il compito non banale, fornendo i risultati che dànno meno informazioni per la classifica. Ma

questo era compito nostro ... Sorprendentemente, nessuno sa come minimizzare il numero di incontri nel caso generale di un numero n qualsiasi di giocatori!

Peccato, perché sarebbe anche il minimo numero di confronti necessari in generale per ordinare n elementi, e le operazioni di ordinamento sono, come sappiamo, assai comuni. C'è ancora molto da scoprire in informatica (e non solo)!

Cocci, la coccinella digitale - al massimo 20 punti

Questa prova va svolta usando il computer. La soluzione deve però essere riportata nel retro della pagina. Per iniziare la prova, cliccate sul bottone "Coccinella".

Lo scopo del gioco è programmare la coccinella Cocci in modo che raggiunga la sua tana. Sul computer è disponibile un video che illustra il funzionamento del gioco. Per non disturbare i vicini tenete il volume audio molto basso o addirittura spento. Il testo di quanto viene detto è sostanzialmente riportato qui:

Cocci è una coccinella digitale, il cui comportamento è determinato dalle istruzioni che potete trascinare nella piccola finestra gohome che vedete a sinistra. A destra avete una finestra con le istruzioni disponibili per il compito assegnato che, di volta in volta, sarà quello di portare Cocci alla sua tana, toccando le stelline eventualmente presenti o seguendo un percorso prefissato? Per far eseguire a Cocci le istruzioni che avete trascinato nella finestra gohome potete cliccare sul punto esclamativo (!), per eseguire una sola volta tutte le istruzioni presenti, oppure sull'orologio, per ripetere le istruzioni indefinitamente, fino a quando, se tutto va bene, Cocci raggiungerà la tana. Il punto esclamativo è molto utile, in modo particolare per vedere l'effetto di una singola istruzione. Esistono anche istruzioni condizionali, i test, con i quali è possibile ottenere effetti complessi, come costringere la coccinella a voltarsi di 180 gradi solo se la sua testa nera (o, con altre istruzioni, l'occhio giallo o quello verde) incontra un oggetto o una zona di colore giallo (o altro), ottenendo quindi questo effetto di rimbalzo rispetto all'oggetto di colore giallo. Altri tasti non vi servono: se li cliccate probabilmente fanno solo confusione; mentre il cerchio sulla sinistra serve per cancellare la finestra gohome: questo vi costringe a ricominciare, quindi sarebbe meglio se non lo usaste.

Il gioco è fatto da 6 livelli, a ciascuno dei quali si ha accesso dopo aver risolto quelli precedenti. **I primi 4 livelli danno un punteggio massimo di 10 punti. Gli ultimi due livelli sono piú difficili e daranno diritto ad altri 10 punti**. Per ogni quesito il punteggio potrà essere ridotto, in relazione alla minore eleganza della soluzione. Saranno ritenute eleganti le soluzioni che: partono dalla situazione iniziale (quella che si ottiene cliccando sul tasto ricomincia) e procedono senza cambiare il contenuto della finestra gohome, usano le istruzioni cosí come sono (ossia senza cambiarne i parametri), toccano le stelline nel campo di gioco, seguono il piú possibile la strada indicata in marrone.

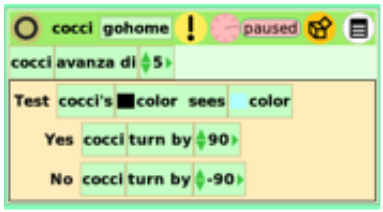
Per ciascun livello dovete indicare la soluzione sul retro di questo foglio, trascrivendo esattamente le istruzioni che avete inserito nella finestrella `gohome` che guida il comportamento della coccinella.

Attenzione: ad ogni nuovo accesso al gioco è necessario ripartire dal primo livello, quindi ricordate di trascrivere le soluzioni man mano che le trovate!

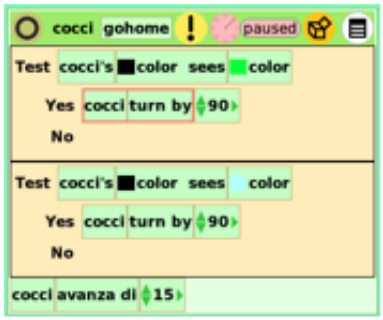
Soluzione Sono possibili soluzioni diverse rispetto a quelle proposte. In particolare si noti che, poiché le istruzioni vengono eseguite *in parallelo*, è possibile elencarle in ordine differente, ottenendo lo stesso risultato.

Quesito 1

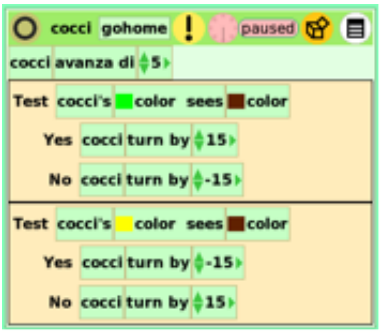
Quesito 2



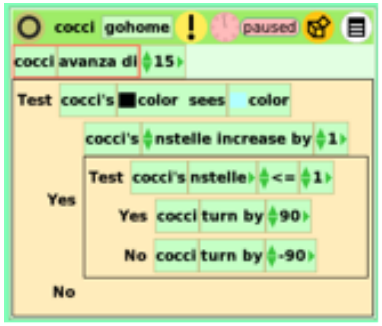
Quesito 3



Quesito 4



Quesito 5



Quesito 6

Elaborazione di testi con Wiki - al massimo 15 punti

Questa prova va svolta sul computer. Per iniziare la prova, cliccate sul bottone "Wiki"

Un **wiki** è un sito Web che può essere modificato dai suoi utilizzatori e i cui contenuti sono sviluppati in collaborazione da tutti coloro che vi hanno accesso. In questa prova vi verrà fornito un testo non formattato (cioè senza impaginazione né stili) che dovrete formattare per riprodurre il piú fedelmente possibile il documento proposto in allegato, utilizzando il wiki che trovate sul vostro computer. La formattazione in un wiki viene ottenuta utilizzando un linguaggio di markup, cioè un insieme di parole (o simboli) speciali, dette marcatori, che vengono inserite nel documento stesso insieme al testo vero e proprio del documento e che specificano i comandi di formattazione. Piú precisamente, per ottenere del testo in un particolare formato (grassetto, elenco, ecc.), occorre inserire il marcatore di quel formato subito prima e subito dopo il testo da formattare. Ad esempio, il marcatore per la sottolineatura è `__` (due caratteri di sottolineatura) e quindi se voglio sottolineare devo scrivere `__sottolineare__`. L'effetto dell'uso dei marcatori non è immediato. Quando si usa un linguaggio di markup per preparare un documento, si accede infatti al documento in due modi alternativi:

- in modalità di *lavoro* si accede al documento come *testo sorgente*, ovvero testo non formattato decorato con marcatori,
- in modalità *visualizzazione* si accede invece al documento formattato.

Qui di seguito trovate tutte le istruzioni che vi servono su come scrivere una pagina wiki.

Accesso e interfaccia

Accedete alla prova "Wiki" cliccando sulla relativa voce nel menu. Nella parte a sfondo giallo troverete un'interfaccia con alcuni bottoni in alto a destra e il testo da formattare. I bottoni sono:

- MODIFICA: cliccate qui per accedere e lavorare al testo da formattare
- INFORMAZIONI: cliccate qui per vedere la storia delle revisioni del documento che avete prodotto Se avete cliccato sul bottone MODIFICA, appariranno, sempre in alto a sinistra, i seguenti bottoni:
  - SALVA MODIFICHE, che serve, come dice il nome, per salvare il lavoro fatto. Ricordatevi di salvare prima che scada la vostra quota di tempo!
  - ANTEPRIMA, che serve per vedere l'effetto dei comandi di formattazione che avete inserito, cioè per vedere come apparirà il vostro documento, ma senza salvarlo; se siete soddisfatti e volete salvare, andate in cima alla pagina e troverete il bottone SALVA MODIFICHE; se invece volete fare altri interventi sul vostro documento prima di salvare, sopra all'anteprima trovate il testo sorgente su cui riprendere a lavorare.

– ANNULLA, che cancella le modifiche fatte dall'ultimo salvataggio e ripristina quindi la versione precedente.

Formattazione per il Wiki

Vediamo qui di seguito i principali tipi di formattazione e i relativi marcatori da utilizzare. Le regole sono descritte attraverso l'uso di esempi: troverete varie tabelle contenenti a sinistra il testo in modalità di lavoro (testo sorgente, su sfondo grigio) e a destra lo stesso testo in modalità visualizzazione.

Stili Nelle pagine wiki si possono utilizzare diversi stili per i caratteri. Ecco i piú utilizzati.

Scrivere:	utilizzando l'apostrofo (') per ottenere:
`''enfasi (corsivo)''`	*enfasi (corsivo)*
`'''grassetto'''`	**grassetto**
`'''''grassetto-corsivo'''''`	***grassetto-corsivo***

Scrivere:	Per ottenere:
`__sottolineato__`	sottolineato
`,,sotto,,scritto (pedice)`	$_{\text{sotto}}$scritto (pedice)
`^sovra^scritto (apice)`	$^{\text{sovra}}$scritto (apice)
`~-piú piccolo-~`	piú piccolo
`~+piú grande+~`	piú grande
`--(cancellato)--`	~~cancellato~~

Per scrivere del testo non formattato, cioè per fare in modo che la formattazione wiki venga ignorata, racchiudere il testo tra coppie di tre parentesi graffe (`{{{` e `}}}`), come nell'esempio.

Il testo:	viene visualizzato:
`{{{testo--non--formattato}}}`	`testo--non--formattato`

Titoli Il marcatore per fare i titoli è il carattere di uguale (`=`). Per creare un titolo, iniziare la riga con un carattere di uguale (`=`) seguito da uno spazio bianco, mettere il titolo, e chiudere la riga con uno spazio bianco seguito da un `=` . Sottotitoli, sottosottotitoli, ecc. vengono ottenuti utilizzando piú caratteri `=` (fino a cinque), in ugual numero prima e dopo il testo del titolo.

Quindi scrivendo	si ottiene:
`= Titolo 1° livello =`	**Titolo 1° livello**
`== Titolo 2° livello ==`	**Titolo 2° livello**
`=== Titolo 3° livello ===`	**Titolo 3° livello**
`==== Titolo 4° livello ====`	**Titolo 4° livello**
`===== Titolo 5° livello =====`	**Titolo 5° livello**

Paragrafi Per andare a capo occorre inserire una o piú righe vuote.:

Il risultato che si ottiene scrivendo:	è:

`Riga1`
`Riga2 sulla stessa riga di Riga1`
 	
`Riga3`
 	
`Riga 4`
	Riga1 Riga2 sulla stessa riga di Riga1
	Riga3
	Riga4

Elenchi È possibile creare degli elenchi in modo molto semplice. Tutto quello che bisogna fare è applicare un rientro di uno spazio alle righe che contengono gli elementi dell'elenco. Per concatenare elenchi su livelli differenti, è necessario utilizzare una diversa profondità di rientro. Tutti gli elementi allo stesso livello di rientro appartengono allo stesso (sotto-) elenco. Si possono avere sia elenchi puntati (non numerati) sia elenchi numerati.

Per creare un elenco puntato utilizzare un asterisco (*). Attenzione: gli spazi che vedete all'inizio delle righe sono necessari!

Per andare a capo all'interno di un elenco, basta inserire <
> alla fine della parola dopo la quale si vuole andare a capo

Scrivendo:	Si ottiene questo elenco:

`* Elemento 1`
`  * Elemento 1.1`
`  * Elemento 1.1.1`
`  * Elemento 1.1.2`
`* Elemento 2`
	• Elemento 1
	• Elemento 1.1
	• Elemento 1.1.1
	• Elemento 1.1.2
	• Elemento 2

Utilizzando 1. al posto dell'asterisco:	Si ottiene un elenco numerato come il seguente:

`1. Elemento 1`
`  1. Elemento 1.1`
`  1. Elemento 1.1.1`
`  1. Elemento 1.1.2`
`1. Elemento 2`
	1. Elemento 1
	1. Elemento 1.1
	1. Elemento 1.1.1
	1. Elemento 1.1.2
	2. Elemento 2

Si possono anche avere elenchi numerati con lettere o numeri romani e mettere insieme piú tipi di elenchi, numerati e non.

Questo esempio:	verrà visualizzato come segue:

`* Romano minuscolo`
`  i. uno`
`  i. due`
`* Romano maiuscolo`
`  I. uno`
`  I. due`
`* Alfabetico minuscolo`
`  a. uno`
`  a. due`
`* Alfabetico maiuscolo`
`  A. uno`
`  A. due`
	• Romano minuscolo
	i. uno
	ii. due
	• Romano maiuscolo
	I. uno
	II. due
	• Alfabetico minuscolo
	a. uno
	b. due
	• Alfabetico maiuscolo
	A. uno
	B. due