



Informatica dilettevole e curiosa

Giochi matematici e giochi informatici –
somiglianze e differenze

Maurizio Codogno – 29 maggio 2013

Esiste una lunghissima tradizione di divulgazione matematica fatta attraverso problemi e giochi. Per quanto riguarda il campo informatico, non solo italiano ma anche anglosassone, è invece molto più difficile trovare approcci. Ci sono varie ragioni per questo, e il problema non è necessariamente legato al fatto che l'informatica è molto più recente come scienza.

Devo ammettere che anch'io, nonostante in fin dei conti abbia un doppio background matematico e informatico, e abbia iniziato a giocare con l'informatica trentacinque anni fa o giù di lì, mi trovo molto più a mio agio con i giochi matematici; però credo di poter comunque dare anche qualche idea sull'informatica, presa da esempi pratici.

Ah: il titolo è preso da quello del libretto di Italo Gherzi, «Matematica dilettevole e curiosa».

I giochi matematici...



... non saranno
vecchi come l'
umanità,
ma poco ci manca!

(papiro di Rhind, 1650 a.
C.)

29 maggio 2013

Informatica dilettevole e curiosa - Maurizio
Codogno

Sembra incredibile, ma i primi esempi di problemi matematici (e della loro soluzione...) che conosciamo sono vecchi di quasi 4000 anni. Il più antico testo a noi noto è il Papiro di Rhind, che risale al 1650 a.C. ma probabilmente è una copia di un testo composto tra il 2000 e il 1800 a.C..

Nel Papiro di Rhind si trovano tabelle di frazioni e 84 problemi aritmetici, algebrici e geometrici con le relative soluzioni.

Oggettivamente non si può ancora parlare di «giochi» matematici, almeno nel senso che daremmo oggi al termine; in effetti sono più che altro problemi pratici, e il papiro è più assimilabile a un libro di testo, con lo scriba che spiega con pazienza come risolvere il problema.

Libri di problemi per divertire (?)



29 maggio 2013

Informatica dilettevole e curiosa - Maurizio
Codogno

In compenso, già dal medioevo si possono trovare raccolte di problemi nate con lo scopo, anche se magari secondario, di convincere il riottoso lettore ad appassionarsi alla materia proponendo dei giochi. Alcuni titoli:

Alcuino di York, *Propositiones ad acuendos juvenes*, ~800 – questo libro nacque come risultato della riforma carolingia dell'insegnamento

Claude Gaspard Bachet, *Problèmes plaisants et délectables, qui se font par nombres*, 1624 – notate che siamo passati alla lingua popolare, e non più al latino. Sennò come si potrebbe giocare?

Luca Pacioli, *De viribus quantitatis*, ~1500 – anche qua, nonostante il titolo latino, il testo è in italiano.

Si parla proprio di problemi “divertenti”: non credo che in quello di lupo, capra e cavolo ci fossero usi

pratici...

Non so poi se sia un caso che tutti e tre questi libri siano stati scritti da ecclesiastici: magari è semplicemente perché loro dovevano pensare all'insegnamento mentre i poeti e scrittori no, o magari è perché avevano più tempo a disposizione.

(immagini: http://en.wikipedia.org/wiki/File:Raban-Maur_Alcuin_Otgar.jpg
<http://books.google.com.ar/books?id=Zy0PAAAAQAAJ&hl=en>
<http://www.uriland.it/matematica/DeViribus/indice1.html>)

Ci sono problemi e problemi!

- Problemi complicati: teorema dei quattro colori
- Problemi irrisolti: P=NP
- Problemi non risolvibili: Halting problem
- Problemi scolastici: disegnare le aste ☺, le tabelline
- Problemi noiosi: una frazione pari a $\frac{1}{2}$ usando tutte e dieci le cifre

29 maggio 2013

Informatica dilettevole e curiosa - Maurizio Codogno

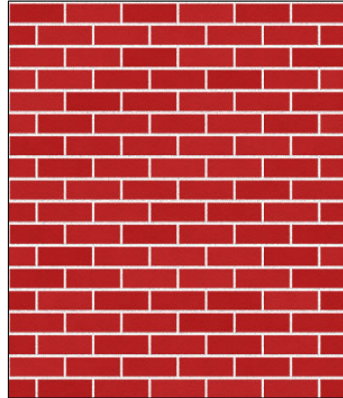
In effetti bisogna anche intendersi su quali problemi si vogliono trattare. In informatica, come del resto in matematica, abbiamo vari tipi di problemi per tutte le salse:

- I **problemi complicati**, la cui risoluzione richiede conoscenze estremamente specialistiche e spesso ad ampio raggio (non è un ossimoro) oppure consistono semplicemente in troppi conti.
- I **problemi irrisolti**, che evidentemente sono ancora peggio: il buonanima di Hilbert diceva “*Wir müssen wissen. Wir werden wissen*” ma è stato sconfessato da Gödel.
- I **problemi non risolvibili**, di cui si può dimostrare la non risolvibilità – ma non si sta certo giocando.
- I **problemi scolastici**, che sono sicuramente semplici, fin troppo semplici, ma proprio per questo non sono interessanti.
- I **problemi noiosi**, o che almeno annoiano me: molti

•dei problemi che si trova(va)no nei libri di giochi matematici richiedono di mettersi lì e fare tanti conti, per trovare la configurazione giusta. Su questo ritornerò però dopo: in fin dei conti è vero che le cose sono un po' diverse ora che i conti noiosi li fa il computer, ma è anche vero che la noia si può comunque spostare a un altro livello. Può esserci una sfida solo se il programma per scriverli è interessante...

Ad ogni buon conto, non parlerò di nessuno di questi tipi di problemi se non degli ultimi, e anche di quelli solo di striscio.

Esercizi o problemi?



29 maggio 2013

Informatica dilettevole e curiosa - Maurizio Codogno

Attenzione, però! c'è una differenza molto grande tra *problemi* ed *esercizi*.

Un esercizio è qualcosa che in linea di principio si sa come risolvere, in genere applicando qualcosa che si è appena studiato. Non è detto che uno riesca a risolverlo, i calcoli sono sempre insidiosi, ma appunto è più che altro una questione di esercizio. Per costruire un muro si fa una fila di mattoni, poi una seconda fila sfalsata, una terza fila direttamente sopra la prima e così via: si aggiungano malta, livella e filo a piombo q.b.

Un problema invece è qualcosa di cui non si sa assolutamente a priori la strada per arrivare alla soluzione: si prova man mano l'una o l'altra via, se possibile sfruttando tutte le conoscenze pregresse, ma essendo alla fine costretti magari a inventarsi

qualcosa di nuovo.

Chiaramente la ripartizione non è sempre così netta: i problemi per i ragazzi delle medie saranno più che altro esercizi, ma ci dovrebbe sempre essere quel qualcosa che fa la differenza. Può essere utile vedere le caratteristiche dei giochi matematici, per avere un'idea di cosa rende interessante un problema. Ci sono due categorie principali: i giochi che si risolvono con un a-ha (= capire qual è l'invariante o il teorema da usare, trovarlo e avere immediatamente la risposta) e quelli essenzialmente enumerativi (= ci sono tante possibilità da provare fino a trovare quella giusta). Come si traspone tutto questo in informatica? Prima di dare una risposta, vediamo due problemi «tipici» (o almeno di quelli che erano tipici cent'anni fa, prima che ci fossero i computer)

(immagini: da <http://oscarferrari.wordpress.com/2007/12/07/il-muro-chiude/> e <http://www.comune.torino.it/deutsch/musei/schede/ma1.htm>)

Giochi a-ha

131. FINDING A SQUARE

Here are six numbers: 4,784,887, 2,494,651, 8,595,087, 1,385,287, 9,042,451, 9,406,087. It is known that three of these numbers added together will form a square. Which are they?

The reader will probably see no other course but rather laborious trial, and yet the answer may be found directly by very simple arithmetic and without any experimental extraction of a square root.

Non serve mica un computer per trovare la soluzione, bastano carta e penna!

29 maggio 2013

Informatica dilettevole e curiosa - Maurizio Codogno

Naturalmente oggi sarebbe possibile risolvere questo problema molto semplicemente: si fanno le 120 possibili somme a tre a tre dei numeri, e si calcola la loro radice quadrata. Un programma che fa questo lo si scrive in fretta, ma non è che ci si divertirebbe molto: diciamo che sarebbe un esercizio.

Cent'anni fa fare tutte quelle radici quadrate era un lavoraccio, e nemmeno le tavole dei logaritmi sarebbero state così utili: però basta calcolare la radice numerica dei sei numeri dati, verificare quali se sommate possono essere radici numeriche di un quadrato e scoprire che uno dei numeri è il secondo, mentre gli altri due sono da scegliere tra il quarto, il quinto e il sesto. Ma se si usasse il quinto, la somma finirebbe per 189 oppure 389, mentre nei quadrati se le ultime due cifre sono 89 la terz'ultima è pari. Quindi la risposta è «il secondo, il quarto, il sesto».

Insomma, un problema abbastanza interessante dal punto di vista dei giochi matematici (anche se non richiede certo conoscenze matematiche di chissà quale livello), ma che non è per nulla informatico.

(problema da Henry Ernest Dudeney, 536 Curious Problems & Puzzles)

Giochi enumerativi

129. DIGITS AND SQUARES

One of Rackbrane's little Christmas puzzles was this: (1) What is the smallest square number, and (2) what is the largest square number that contains all the ten digits (1 to 9 and 0) once, and once only?

La calma è la virtù dei computer:

```
map { $n=$_*$_; print "$n\n" if  
      grep($n =~ /$_/, 0..9) == 10; }
```

$$32043^2 = 1026753849$$

$$99066^2 = 9814072356$$

29 maggio 2013

Informatica dilettevole e curiosa - Maurizio
Codogno

In questo caso, invece, non ci sono molti trucchi che si possono utilizzare, se non partire a fare quadrati da 31623 in su e da 99999 in giù e al limite saltare quelli che finiscono per zero. Insomma, un problema che dal punto di vista matematico e dei giochi matematici è noioso perché ci sono solo da fare conti e non c'è nulla di interessante; e dal punto di vista informatico non è chissà che. Non confondiamo insomma l'implementazione pratica (il programma) con la teoria che sta sotto: sarebbe come dire che la matematica è semplicemente sapere fare conti, il che non mi pare sia proprio il caso!

Insomma, ci vuole qualcosa di diverso. Ma prima di trovarlo, c'è ancora qualcosa da dire su come si fanno i problemi.

Giochi prettamente informatici

Nascono con l'informatica: prima non se ne sentiva la necessità! (o no?) Esempi:

- Algoritmi / diagrammi di flusso
- Logica (binaria)
- Backtracking
- Operazioni su stringhe
- Ricorsione
- Strutture dati
- Alberi e grafi

29 maggio 2013

Informatica dilettevole e curiosa - Maurizio
Codogno

L'informatica è molto più recente della matematica: ha meno di cent'anni. È abbastanza chiaro che non ci può essere una grande teoria sui giochi informatici, visto che non abbiamo 4000 anni di storia a disposizione: ma questo non significa che non si possano avere delle categorie di giochi che hanno come base le operazioni prettamente informatiche. Non c'è nulla di strano che nelle categorie qui elencate troviamo giochi «matematici»: in fin dei conti, è la stessa cosa per cui i problemi di fisica si risolvono con tecniche matematiche, come dicevo nella slide precedente.

Scopriremo poi come alcune di queste categorie siano state «usurpate» dalla matematica, nel senso che ci sono vecchi giochi matematici che in realtà possiamo definire ex post come giochi informatici.

Algoritmi / diagrammi di flusso

“perché in una ricetta di cucina ogni tanto troviamo scritto q.b.?”



29 maggio 2013

Informatica dilettevole e curiosa - Maurizio Codogno

La prima cosa da sapere è che il problema del lupo, della capra e del cavolo ha più di 1000 anni, ed era nel libro di Alcuino:

XVIII. PROPOSITIO DE HOMINE et CAPRA et LUPO

Homo quidam debebat ultra fluvium transferre lupum, capram, et fasciculum cauli. Et non potuit aliam navem invenire, nisi quae duos tantum ex ipsis ferre valebat. Praeceptum itaque ei fuerat, ut omnia haec ultra illaesa omnino transferret. Dicat, qui potest, quomodo eis illaesis transire potuit.

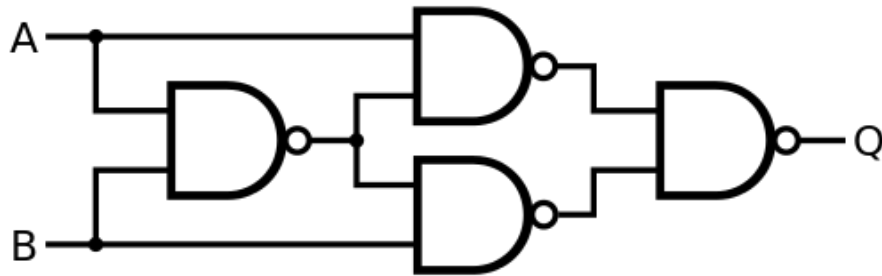
La seconda cosa da sapere è che questo problema di matematico non ha NULLA. È davvero un puro problema informatico, nel senso che occorre trovare un algoritmo per la soluzione.

La terza cosa da sapere è che anche una ricetta sarebbe un algoritmo, se non fosse per i famosi q.b. che si trovano fin troppo spesso (insieme a tanti altri giudizi soggettivi...)

(immagine da A.M. Kordemsky, *The Moskow Puzzles*)

Logica (binaria)

“i mattoni più semplici”



29 maggio 2013

Informatica dilettevole e curiosa - Maurizio Codogno

Anche la logica era nata come parte della filosofia, è stata assorbita dalla matematica da Frege in poi, ma ora è probabilmente più informatica. Le tabelle di verità sono ovviamente alla base dei calcolatori, e i costrutti `if..then..else` sono in fin dei conti applicazioni della logica.

Costruire le varie porte logiche partendo dalla sola NAND è l'esempio quintessenziale (nella figura si mostra come ottenere uno XOR); ma anche la soluzione dei soriti di Lewis Carroll, o se preferite delle Prove d'intelligenza della Settimana Enigmistica, sono sempre esempi di come usare la logica nella definizione di algoritmi.

(immagine da http://en.wikipedia.org/wiki/File:XOR_from_NAND.svg)

Backtracking

“se tutte le strade portano a Roma, da Roma **possiamo** andare dappertutto!”



29 maggio 2013

Informatica dilettevole e curiosa - Maurizio Codogno

Se si nota che la **penultima mossa** per risolvere un problema non è poi così difficile, può avere senso provare a risolvere il problema dal fondo. C'è tutta una teoria al riguardo: Raymond Smullyan ha scritto *The Chess Mysteries of Sherlock Holmes* dove i problemi non sono affatto classici e spesso richiedono di scoprire quali sono state le mosse che hanno portato alla configurazione della scacchiera.

Nell'informatica il backtracking è fondamentale, anche se generalmente non è il mezzo più veloce per risolvere un problema; il concetto informatico è però un po' diverso da quello del backtrack nei giochi. Nel caso informatico l'esempio quintessenziale è quello delle otto regine: quando non si riesce più a posizionare una regina, si torna indietro di un passo e si ricomincia da lì, mentre nel gioco il backtracking è più che altro un partire dal fondo e tornare indietro. C'

è sempre l'idea di «tornare indietro», però.

Esempio:

- Schede rosse e verdi: Ci sono 42 schede rosse e 24 verdi, e si estraggono schede finché sono dello stesso colore; quando se ne pesca una di colore diverso la si rimette dentro e si ricomincia da capo. (Notate che si può tornare ad avere una carta dello stesso colore di prima) Qual è la probabilità di finire con una scheda rossa? Ecco, non mettetevi a provare tutte le configurazioni 😊 Se mettiamo in fila tutte le carte nell'ordine in cui sono uscite, vediamo che l'ultimo gruppo di carte è di un colore, e il penultimo deve essere del colore opposto. (Sul terz'ultimo gruppo non sappiamo nulla; immaginiamo di avere due carte rosse e due verdi, potremmo avere RV, RV, VV che dà R R VV oppure RV, VR, RV, V che dà R V R V). Visto che è indifferente cambiare l'ordine dei due ultimi gruppi, la probabilità di avere un colore oppure l'altro per ultimo è la stessa.

Operazioni su stringhe

“i mattoncini un po' meno logici”

1	$MI \in \Omega$
11	$xI \in \Omega \Rightarrow xIU \in \Omega$
21	$Mx \in \Omega \Rightarrow Mxx \in \Omega$
1112	$III \Rightarrow U$
3112	$UU \Rightarrow ()$
211213	
312213	
...	
	$MU ?$

29 maggio 2013

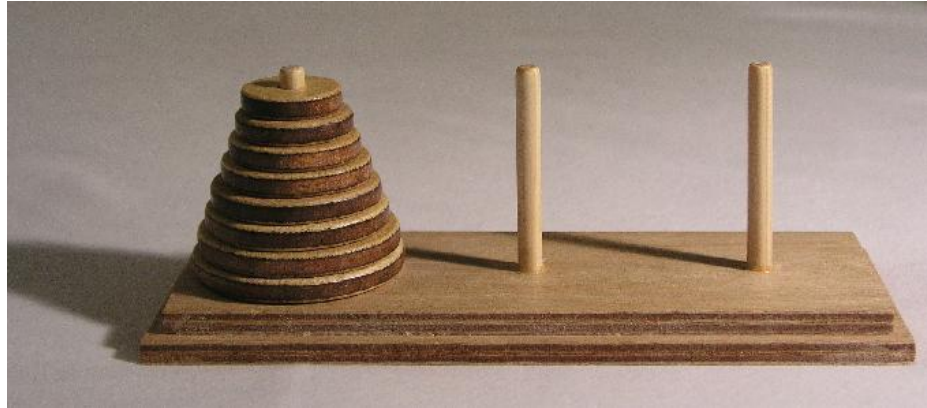
Informatica dilettevole e curiosa - Maurizio
Codogno

L'esempio che vedete qui sopra NON È quello che siete sicuramente abituati a vedere, ma è molto simile. In un certo senso qui stiamo parlando di strutture dat: semplicemente invece che leggerla da sinistra a destra si compattano i vari numeri.

La parte di destra è il sistema MIU di Hofstadter: una struttura logica con certe regole di inferenza, che permettono di creare «teoremi». È possibile ottenere MU da quelle regole?

Ricorsione

“ricorsione, s.f.: vedi *ricorsione*”



29 maggio 2013

Informatica dilettevole e curiosa - Maurizio Codogno

La ricorsione (che non è mai vera ricorsione, a differenza di quanto ho scritto nel teaser: non si arriva mica all'infinito!) è un altro dei concetti chiave dell'informatica... e dei giochi matematici.

La Torre di Hanoi, per esempio, inventata da Edouard Lucas nella seconda metà del XIX secolo, è il perfetto esempio di problema ricorsivo: se sai risolvere la torre per $N-1$ pezzi, sai anche farlo per N .

(immagine da https://it.wikipedia.org/wiki/File: Tower_of_Hanoi.jpeg)

Strutture dati

“chi scrive un linguaggio che prevede il valore

0 but true

dovrebbe farsi vedere da uno buono”

a = a+b

b = a-b

a = a-b

29 maggio 2013

Informatica dilettevole e curiosa - Maurizio
Codogno

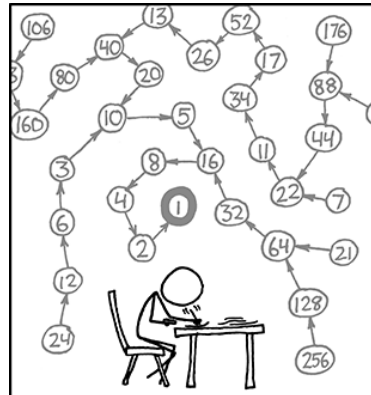
Larry Wall è sempre stato un tipo molto pittoresco, e Perl ne è un esempio preclaro: «0 but true» vale zero se calcoli il valore come numero, ma true se lo calcoli come booleano. (Ho poi scoperto che nei perl moderni c'è la possibilità di definire una variabile scalare di tipo dualvar, con un valore numerico e uno di stringa: cosa dicevo di Larry Wall?)

Purtroppo, non sono mai riuscito a trovare un esempio di gioco informatico che prevedesse l'uso serio di strutture dati. L'esempio che ho messo nella slide, che è la soluzione al problema «scambiate il valore di due variabili intere senza usare una terza variabile di appoggio» è infatti informaticamente **sbagliato**, come dovrete sapere tutti: se a e b sono molto vicini al massimo intero esprimibile, a+b è fuori scala.

Forse si potrebbe immaginare qualche problema in cui si impacchettano valori in maniera più o meno interessante: magari prima o poi ne invento uno!

Alberi e grafi

“gira che ti rigira, ti farò tornare qui prima o poi!”



29 maggio 2013

Informatica dilettevole e curiosa - Maurizio Codogno

Esempi di problemi informatici che richiedono la teoria dei grafi ce ne sono molti: pensate per esempio alla ricerca di un cammino hamiltoniano su un solido platonico. In questo caso preferisco parlare di un' applicazione un po' diversa (no, non vi chiedo di dimostrare la congettura di Collatz, anche perché – ammesso che ci riusciste e diventaste immediatamente famosi – avreste fatto matematica!)

Lo **spazio degli stati** è un concetto della fisica, ma a volte anche un matematico o un informatico può usarlo. Se si va sempre da uno stato a un altro e il numero complessivo di stati è finito allora prima o poi si ritorna a uno stato già visto; se preferite.

Attenzione, però: se si vuole dimostrare che si torna allo stato iniziale, occorre mostrare che non ci sono cicli, altrimenti si potrebbe finire attirati lì, come nel caso del 6174 (si prenda un numero, lo si scriva prima

con le cifre in ordine decrescente e poi in ordine crescente, e si faccia la sottrazione, continuando così...)

Esempio:

L'isola di Triabol, con tutti incroci a Y. Se uno gira sempre alternativamente a destra e a sinistra, prima o poi tornerà al punto di partenza. Qui lo spazio degli stati è la tripletta (incrocio, strada da dove si è arrivati, ultima svolta fatta) e il trucco è che da ogni tripletta si può andare solo a un'altra tripletta, quindi si fa un "giro delle triplette", anche se magari non le si tocca tutte.

(immagine: <http://xkcd.com/710/>)



Giochi matematici o informatici?

Il fatto che li si consideri matematici non significa che non siano essenzialmente informatici!

- Combinatoria
- Strategia ingorda

29 maggio 2013

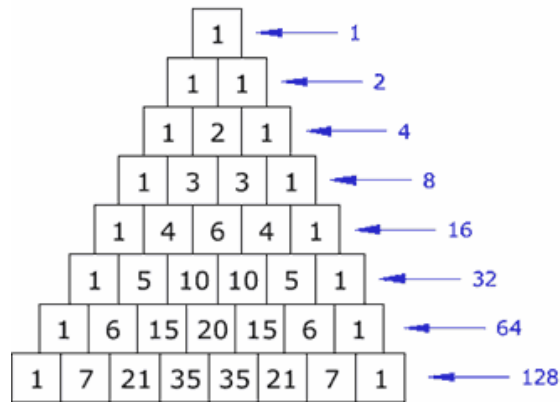
Informatica dilettevole e curiosa - Maurizio Codogno

Non butterei via del tutto i problemi matematici non informatici in senso stretto, però.

Certo, per i problemi di questa categoria la parte più prettamente informatica è quella di preparare un algoritmo (costruttivo) per trovare la soluzione, e quindi come dicevo all'inizio la cosa è un po' meno teorica e più pratica, però tanto vale provarci.

Combinatoria

“uno, due, tre, stella!”



29 maggio 2013

Informatica dilettevole e curiosa - Maurizio Codogno

Dal Superenalotto in giù, le formule **combinatorie** (sì, i matematici dicono così, per colpa dell'inglese) sono spesso molto utili, anche se in effetti nelle applicazioni pratiche occorre fare parecchi conti che usano numeri enormi. Se volete un esempio di combinatoria all'opera, potete prendere il triangolo di Tartaglia: ci potete trovare di tutto.

Simile alla combinatoria almeno dal punto di vista logico è anche l'operazione di **contare**. Siete capaci tutti a contare, no?

- Taglia e mangia: si ha una tavoletta di cioccolato a quadretti di dimensioni mn , e a ogni mossa si può dividere una parte in due oppure mangiare un singolo quadretto se è già stato isolato. Non importa qual è l'ordine delle mosse fatte: ogni taglio aumenta di uno il numero di parti e ogni mangiata lo riduce di uno,

•quindi ri assemblando l'ordine delle mosse si può contare il numero di mosse totali che sarà $2mn-1$

(immagine: da <http://www.mathsisfun.com/pascals-triangle.html>)

Strategia ingorda

“ogni tanto è meglio fare le cose con calma”



29 maggio 2013

Informatica dilettevole e curiosa - Maurizio Codogno

La “greedy strategy”, che cerca sempre di fare il maggior passo possibile, è comoda, ed è quella che in informatica viene usata più spesso perché è la più facile da implementare: ma a volte può portare a non ottenere il risultato voluto. Il guaio è che con questa strategia a volte ci si preclude una via un po’ più tortuosa inizialmente ma che alla fine porta a risultati migliori (e qui si ritorna a parlare di backtracking).

Esempio:

- John, Paul, George e Ringo devono attraversare il ponte sul fiume Mersey al buio, e hanno una sola lampada a disposizione. Il ponte regge al più due persone: visto che hanno bevuto molto e sono malfermi sulle gambe, per percorrerlo ci mettono rispettivamente 1, 2, 5, 10 minuti. Possono farcela in 17 minuti? Se si parte dal principio ingordo, che cioè sia John a portare tutti dall’altra parte uno per volta

- perché è quello che può tornare indietro più facilmente, no. Il trucco è mandare insieme a Ringo il più lento degli altri, cioè George: così almeno lui non perde tempo inutile.

(immagine: http://www.hvrsd.org/timberlane/students2011/CurtisH/images/Smurfs_Color_Pictures_Greedy_Smurfs.jpg)

Cui prodest?

Perché bisogna perdere tempo con i giochi (informatici o matematici che siano?)

- Perché sono comunque più divertenti che studiare la teoria
- Perché rafforzano le capacità logiche
- Perché servono in pratica

29 maggio 2013

Informatica dilettevole e curiosa - Maurizio Codogno

Io mi sento già dire tante volte «ma a che cosa serve la matematica? Basta al più saper far di conto!» Immaginate la faccia che fa la gente con i GIOCHI matematici, che sembrano loro più che altro una tortura. Poi magari loro si infliggono altri tipi di torture e li chiamano divertimenti, ma tant'è. Nella slid evi do alcune risposte alla domanda «perché si fanno i giochi matematici e informatici», ma la vera risposta è 42... no, scusate: la vera risposta è che chi fa questi giochi è biecamente competitivo.

Un caso reale

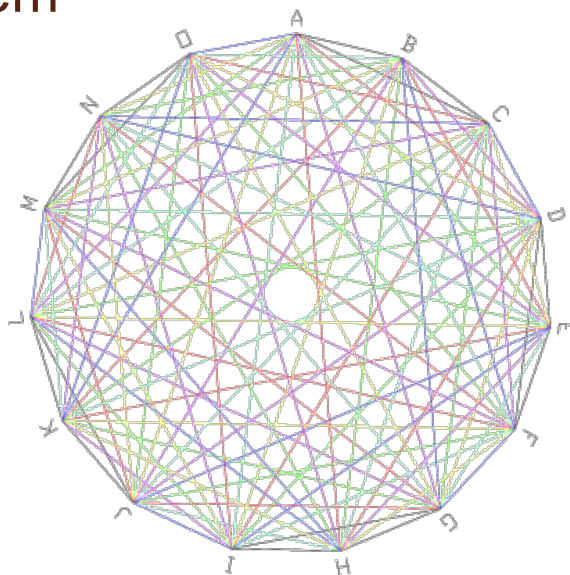
Mi scrive un amico chiedendomi «Ho un certo numero di persone, e devo creare tanti terzetti di persone dove due qualsiasi terzetti hanno al più un elemento in comune. Ho fatto qualche simulazione, ma i risultati variano a seconda del punto di partenza. Puoi darmi una mano?»

29 maggio 2013

Informatica dilettevole e curiosa - Maurizio
Codogno

La storia indicata in questa slide è verissima, e mi è capitata un paio di settimane fa. Notate come il mio amico sia un informatico, ma non un giocatore ☹, e che abbia applicato una greedy strategy: in pratica faceva terzetti finché poteva, e quando non ce la faceva più sceglieva un nuovo compagno a caso (ecco perché i risultati variavano, nel caso ve lo chiedeste)

L'idea: Kirkman's Schoolgirl Problem



29 maggio 2013

Informatica dilettevole e curiosa - Maurizio Codogno

Essendo invece io un giocatore, ho immediatamente ricollegato il problema a quello delle quindici scolarette, che devono essere messe in cinque gruppi da tre per sette giorni la settimana, senza che nessuna coppia si trovi insieme per due giorni. Il problema lo conoscevo dai libri di Dudeney: una rapida guglata mi ha fatto scoprire che in letteratura si chiama Kirkman's Schoolgirls Problem, e che è un caso particolare delle triple di Steiner: la teoria per costruire le triple ottimali l'ho infine trovata a <http://mathoverflow.net/questions/72084/constructing-steiner-triple-systems-algorithmically> dopo aver chiesto in giro *io* con la domanda più corretta. In pratica, l'aver giocato spesso mi ha aiutato a trovare la risposta!

(immagine da http://mathworld.wolfram.com/images/eps-gif/KirkmansSchoolgirlProblem_600.gif)

Siti di problemi

Kangourou ☺ : <http://kangourou.di.unimi.it/>

Project Euler: <http://projecteuler.net/>

UVa Online Judge: <http://uva.onlinejudge.org/>

MindCipher: <http://www.mindcipher.com/>

Computer Science Stack Exchange:
<http://cstheory.stackexchange.com/questions/tagged/puzzles>

Google Code Jam: <http://code.google.com/codejam/contests.html>

Li conoscete meglio di me! Però sono pigro, lascio questi link per chi si trova sullo schermo queste slide. Tenete però conto che molti di questi siti sono più legati alla programmazione vera e propria, che come ben sappiamo è solo una parte dell'informatica.

I Kangourou dell'informatica - <http://kangourou.di.unimi.it/> - sono gestiti qui a Unimi, e sono dedicati ai ragazzi: la programmazione è relativamente secondaria, ci sono giochi più teorici

Project Euler - <http://projecteuler.net/> - è molto legato al lavoro di squadra. Molti di quei problemi sono assolutamente intrattabili!

UVa Online Judge - <http://uva.onlinejudge.org/> - sito dell'università di Valladolid, penso che sia il primo ad avere creato i set di problemi informatici

MindCipher - <http://www.mindcipher.com/> - sito con problemi più o meno informatici

Computer Science Stack Exchange: <http://cstheory.stackexchange.com/questions/tagged/puzzles> - del network Stack Exchange, il sito è dedicato in genere all'informatica teorica, ma ci sono anche giochi informatici segnalati col tag «puzzles».

Google Code Jam: <http://code.google.com/codejam/contests.html> - Big G c'è sempre!