Solucions del primer concurs d'entrenament

Olimpíada Informàtica Catalana 2018

Problema 1: Cursa ciclista

La resposta és Aina i es pot veure fàcilment. Una possible manera és descartant la resta de candidates:

- La Berta arriba després de la Clara, per tant es descarta com a guanyadora.
- La Deva arriba entre l'Aina i la Clara, per tant es descarta.
- La Clara no pot ser, perquè immediatament darrere arriba la Berta, el que implica que la Deva arriba davant d'ella.

En particular, tenim que l'ordre final és, en ordre d'arribada, Aina, Deva, Clara i Berta.

Problema 2: Pere el jardiner

Solució matemàtica

El problema es pot resoldre fent servir nocions bàsiques de combinatòria. Fem una petita repassada per a qui no conegui el concepte.

Suposem que tenim n elements i en volem triar k, de quantes formes ho podem fer? Això es denota com $\binom{n}{k}$ i es llegeix "n sobre k". Us presentem dues maneres diferents de calcular-lo.

- Fórmula: Suposem que anem agafant els elements un a un. Pel primer hi ha n candidats, pel segon $n-1$, ... i per l'últim $n-k+1$. Això dóna $n(n-1)\dots(n-k+1)$ maneres. Però com no ens importa l'ordre, hem de dividir per les maneres d'ordenar k elements, que són $k!$. Per tant obtenim

$$\binom{n}{k} = \frac{n(n-1)\dots(n-k+1)}{k!} = \frac{n!}{k!(n-k)!}$$

- Recurréncia: Fixem-nos en el primer element. Podem o bé agafar-lo o bé no agafar-lo. Si l'agafem, haurem d'agafar $k-1$ elements dels $n-1$ restants. Si no l'agafem, haurem d'agafar k dels $n-1$ restants. Això ens permet escriure la següent recurrència, que és sovint la manera més simple de calcular-ho programàticament:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

  I els casos base són

$$\binom{n}{n} = \binom{n}{0} = 1$$

  Els nombres que surten sovint s'agrupen en forma de triangle, anomenat triangle de Pascal:

n							
0	1						
1	1	1					
2	1	2	1				
3	1	3	3	1			
4	1	4	6	4	1		
5	1	5	10	10	5	1	
6	1	6	15	20	15	6	1
	0	1	2	3	4	5	6
				k			

Ara doncs tenim les eines necessàries per resoldre el problema. Primer triem els dos forats que quedaran lliures, que ho podem fer $\binom{10}{2}$ maneres. Dels 8 forats restants, triem 4 pels tarongers (i això fixa els llimoners també), que ho podem fer de $\binom{8}{4}$ maneres. Per tant, el resultat buscat és $\binom{10}{2}\binom{8}{4} = 45 \cdot 70 = 3150$.

Solució amb programació

El problema també es pot resoldre amb un programa exhaustiu que provi totes les possibilitats. Per exemple, podem posar 10 bucles anidats on decidim què es posa a cada forat i si es una opció vàlida es considera. Cal tenir cura d'evitar comptar repetits. Podeu mirar el codi `pb2.cpp` per veure una possible implementació.

Problema 3: Quadrats perfectes

Solució amb programació

Com que els candidats són pocs, podem començar per 9999 i anar decrementant fins trobar un número que compleixi el que es demana. Podeu mirar el codi `pb3.cpp` per veure una possible implementació.

Solució a mà

Aquesta opció és força farragosa, ja que bàsicament és fer el mateix que la opció anterior manualment, però evitant càlculs inútils. Primer fem una llista dels quadrats perfectes de 3 dígits: des de $31 \cdot 31 = 961$, fins a $10 \cdot 10 = 100$. Després considerem els quadrats amb 4 dígits de gran a petit:

$99 \cdot 99 = 9801 \to 980$ que no està a la llista.
$98 \cdot 98 = 9604 \to 960$ que no està a la llista.
...
$57 \cdot 57 = 3249 \to 324$ que sí està a la llista $(18 \cdot 18)$.

Per tant 3249 és la resposta.

Problema 4: Senyera

Problema pensat per practicar els problemes gràfics, podeu veure una possible solució a `pb4.py`.

Problema 5: Triangles creixents

Hi ha dues parts en aquest problema: trobar les mides de la imatge i pintar els n triangles. Per trobar les mides, l'alçada serà necessàriament $2n$ i l'amplada serà la suma de les bases dels triangles. Després per anar pintant els triangles cal anar iterant sobre cadascun i no equivocar-se amb les coordenades. Podeu trobar una solució a `pb5.py`.

Problema 6: Temperatures

Problema bàsic sobre condicions. Només cal seguir l'enunciat i provar els casos de prova, que són bastant exhaustius. Podeu trobar una solució possible a `pb6.cpp`.

Problema 7: Primers nombres

Problema bàsic sobre bucles. Per veure una possible implementació podeu mirar `pb7.cpp`.

Problema 8: Suma de dígits

La clau del problema és veure com extreure els dígits del nombre. Es pot llegir com a `string` i anar iterant o bé com a enter i anar fent módul 10 (`%`). Vegeu `pb8.cpp` com a possible implementació.

Problema 9: Múltiple més petit

Una possible primera idea seria anar iterant pels múltiples de b, però aquesta solució és massa lenta i no passaria els casos privats. Cal doncs trobar una fórmula.

Si b divideix a ($a \ \% \ b == 0$), llavors la resposta és a. Altrament, el primer múltiple més gran que a serà $(\lfloor \frac{a}{b} \rfloor + 1)b$, és a dir, el resultat de la divisió entera més u, multiplicat per b.

Problema 10: Més Sudokus

Aquest problema es pot resoldre fent *backtracking* per generar tots els possibles sudokus. Perquè el programa sigui prou ràpid, a cada nivell de la recursió recordarem per a cada fila, columna i quadrat quins números han aparegut. Així, només provarem de posar cada cop números que siguin realment vàlids amb aquesta informació, de forma que quan acabem d'omplir sabrem que el sudoku resultant és correcte. Podeu trobar una solució a `pb10.cpp`.