

Solucions Final OICat 2020

Olimpíada Informàtica Catalana

Problema C1: Febre.

Autor: Alex Alvarez

No cal explicar-lo, oi? Amb saber un bàsic de programació i no tenir febre...

Problema Q1: Anys antipàtics.

Autor: Víctor Martín

No és difícil veure que els anys antipàtics són 2020, 2121, 2222, \dots , 2929, i per a calcular la suma no cal ni un programa. Ara bé, pot ser un bon moment per a que comproveu que si els nombres a_1, \dots, a_n estan en progressió aritmètica (és a dir, la diferència entre dos consecutius és sempre la mateixa), llavors aquesta suma és $\left(\frac{a_1+a_n}{2}\right)n$.

Problema C2: Test de Cooper.

Autor: Alex Alvarez

Simplement aneu llegint els n nombres de la llista i aneu actualitzant el màxim que heu trobat. Fixeu-vos que abans de llegir el primer nombre podeu considerar que el màxim trobat és 0, ja que al llegir-lo passarà a valdre el valor llegit.

Problema C3: Senars alternats.

Autor: Víctor Martín

Provant alguns casos petits és veu ràpidament que la suma és n si n és senar, i $-n$, si n és parell. Aquí una possible demostració pel cas parell: Agrupem els n primers elements de dos en dos: $(1, -3), (5, -7), \dots$, i la suma de cadascun d'aquests parells és -2 , així que la suma total és $-2 \cdot \frac{n}{2} = -n$.

Challenges: Podríeu demostrar-ho (amb rigor) pel cas senar? I podríeu demostrar-ho per inducció? I d'alguna manera diferent?

Problema G1: Demostrant Pitàgores.

Autor: Víctor Martín

Oi que els teoremes són molt millors quan tens una demostració? En aquest problema cal anar únicament amb compte de no posar píxels de menys o més. A partir d'aquí, no hi ha cap dificultat.

Problema Q2: Respostes animades.

Autor: Max Balsells

Cal afegir un nombre a de e's a la primera vocal, b a la segona i c a la tercera, amb $a, b, c \geq 0$. Ara cal veure que la longitud total que ens queda no passa de 20 caràcters. Això ho podem fer amb 3 bucles, un per a cada lletra, i comprovant les condicions per a cada possibilitat. I recordeu de descomptar el cas $a = b = c = 0$.

Challenge: La resposta és precisament $\binom{15}{3} - 1$. Això no és cap casualitat. Podríeu veure per què és així?

Problema G2: Simulant temperatures.

Autor: Víctor Martín

Un altre problema d'implementació, cal seguir tot al peu de la lletra. I recordeu que la divisió entera a python es fa amb `//`! Aquest problema està inspirat en [aquest vídeo](#) de 3Blue1Brown. Pot ser un exercici divertit provar com evoluciona la barra sota diferents configuracions inicials, com visualitzar el problema d'altres maneres, o com millorar el model de problema per a fer-lo més realista (el vídeo us donarà algunes idees).

Problema Q3: Últim dígit.

Autor: Izan Beltrán

Provant algun cas més, podeu veure que a partir de $n = 2$, tots els a_n acaben en 7.

Challenge: Podeu demostrar per què? Podeu fer-ho tant mitjançant la fórmula donada com usant propietats cícliques de les potències mòdul un cert nombre m . En aquest segon cas us pot servir demostrar que $a_n = 2^{2^n} + 1$. I si no sabeu de què va això és un bon moment per aprendre'n :)

Problema C4: Perdent els papers.

Autor: Edgar Moreno

Per a cada persona ens guardem la seva posició inicial en un map (a C++) o en un diccionari (a Python). A partir d'aquí, a qui és avançat li augmentem en 1 la seva posició, i al que avança li restem 1. Finalment, recuperem les posicions del map o del diccionari, posant cada nom en la seva posició correcta en un array o vector, i escrivim els noms en aquest ordre.

Problema C5: Explorant un mapa.

Autor: Salvador Roura

Aquest problema és pot fer molt fàcilment si coneixeu algun algoritme de cerca en grafs, com DFS o BFS (el primer és més senzill d'implementar). En cas que no el conegueu, és un bon moment per mirar-s'ho amb calma.

El vostre DFS ha d'anar en compte de no explorar obstacles ni caselles ja explorades, i heu de recordar de marcar com a explorada cada casella que exploreu.

Problema G3: Permutacions de cercles.

Autor: Salvador Roura

De manera semblant al problema anterior, si sabeu què és un backtracking el problema es convertia en força senzill, i igual que abans, si no sabeu com va, és un bon moment per a mirar-ho. Simplement, a cada pas del backtracking, afegim un dels colors no usats, i avancem recursivament fins que els usem tots.

En aquest cas, a més, podem solucionar el problema sense backtrackings, ja que el fet que l'entrada i la sortida són molt limitades, us permetia haver fet cada cas a mà, o fins i tot us podríeu haver precalculat totes les possibles respostes.

Problema Q4: Nombres raonables.

Autor: Víctor Martín

Si un nombre n té k dígit, clarament $f(n) \leq 9^2 \cdot k$, i d'aquí és fàcil deduir que cap nombre amb 4 dígit o més pot ser raonable. A partir d'aquí, proveu tots els nombres n entre 1 i 999, i amb 1000 iteracions per a cadascun d'ells és suficient (veieu per què?). Aquestes iteracions consisteixen en fer $n = f(n)$, si després d'unes quantes arribeu al nombre inicial, llavors aquest és raonable.

Problema C6: Rajoles en fila.

Autor: Maria Prat

Més o menys idèntic al problema del classificatori, però amb alguna petita adaptació pel fet que ara cada color té un pes diferent.

Challenge: Podríeu demostrar, seguint l'explicació de la versió del classificatori, que $f(i) + g(i) = (v - b)|B| + nb$, on $|B|$ és el nombre de rajoles blanques, i d'aquí treure una solució que no usi vectors i usant molt poques variables? Teniu l'spoiler a `C6_sense_vectors.cc`.

Problema C7: Seqüència monòtona.

Autor: Xavier Povill

Donats r i s , siguin A_0, \dots, A_{r-1} seqüències de s elements definides per $A_k = (sk + s, sk + (s - 1), \dots, sk + 2, sk + 1)$. La concatenació de les A_k 's, en l'ordre donat, és una solució vàlida.

Problema G4: Fractals booleans.

Autor: Víctor Martín

El codi que trobareu a Github servirà de millor explicació que el que es pugui posar aquí...

Problema C8: OICat per equips.

Autor: Felix Moreno

Agafem tres vectors, cadascun d'ells amb el participants ordenats per una de les habilitats, de menor a major. Fixeu-vos que si el millor participant de cada habilitat és diferent, ja hem acabat. Si no, un dels participants és el millor en almenys dos categories. En aquest cas, aquest jugador no pot formar part de cap equip balancejat, ja que si en formés part, trencaria la condició que ha de ser el millor en només un dels tres tipus de problemes. Per tant podem descartar el jugador.

Ara hem de repetir el problema, però sense aquest jugador en concret. De nou, o bé trobarem un equip balancejat o haurem de descartar un altre jugador. I anem repetint fins trobar l'equip o fins descartar a tothom.

A nivell d'implementació, podeu guardar un vector de booleans per a dir qui està descartat, i podeu descartar jugadors només eliminant l'últim element d'alguns dels tres vectors d'habilitats.

Fixeu-vos que en el cas de només borrar-lo en dos categories, ens podrà aparèixer un altre cop en l'altra categoria més endavant, i mirant el seu estat al vector de booleans, decidirem si cal eliminar-lo, en cas que hagi estat descartat, o si no.

Problema C9: Partida de dòmino.

Autor: Salvador Roura

Problema "fàcil", però llarg. Les dues dificultats principals d'aquest problema (sense tenir en compte el codi llarg i segurament ple de bugs que quedarà al primer intent) són, en primer lloc, la complexitat del problema, en el sentit que hi estan passant moltes coses. Per això, cal planejar primer com organitzar el codi i quines variables i funcions posar per a simplificar al màxim i evitar duplicacions. La segona complicació està en ordenar d'una manera raonable els dominos. Al codi de github trobareu alguna suggerència. També us hi donem una solució usant classes (els structs són classes on tot és públic per defecte) per si us és d'interès.