

Solucions Final OICat 2020

Olimpíada Informàtica Catalana

Problema C1: Febre.

Autor: Alex Alvarez

No cal explicar-lo, oi? N'hi ha prou amb saber programar una mica i no tenir febre...

Problema Q1: Anys antipàtics.

Autor: Víctor Martín

És fàcil veure que els anys antipàtics són 2020, 2121, 2222, ..., 2929, i per a calcular aquesta 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, llegim els n nombres de la llista, actualitzant a cada pas el màxim que heu trobat. Fixeu-vos que abans de llegir el primer nombre podeu considerar que el màxim trobat és 0, ja que en llegir-lo passarà a valer aquest primer nombre.

Problema C3: Senars alternats.

Autor: Víctor Martín

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

Reptes: Podríeu demostrar-ho (amb rigor) per al 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 tenim una demostració? En aquest problema cal anar únicament amb compte de no posar píxels de menys o de més. A part d'això, no té 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. Per tant, això ho podem solucionar, per exemple, fent 3 bucles, un per a la longitud de cada lletra, i comprovant que la paraula que ens queda no és massa llarga. Recordeu també descomptar el cas $a = b = c = 0$.

Repte: 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

Aquest és un altre problema d'implementació, on bàsicament cal seguir tot el que indica l'enunciat 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 del 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.

Repte: 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 1 a la seva posició, i al que avança li restem 1. Finalment, recuperem les posicions del map o del diccionari, posant cada nom a 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 es resol fàcilment amb algun algorisme de recorregut en grafs, com DFS o BFS (el primer és més senzill d'implementar). En cas que no els conegueu, és un bon moment per estudiar-ho.

El vostre recorregut ha d'anar amb compte de no visitar obstacles, ni tornar a explorar caselles ja explorades. Per això, heu de recordar marcar les caselles ja explorades.

Problema G3: Permutacions de cercles.

Autor: Salvador Roura

Com el problema anterior, el problema és senzill si coneixeu una altra tècnica fonamental, la tornada enrera (*backtracking*, en anglès). A cada pas del backtracking escollim un dels colors encara no usats, i avancem recursivament fins que els hàgim usat tots.

Aquest problema fins i tot es podia resoldre precalculant totes les possibles respostes, ja que el conjunt d'entrades i sortides possibles era molt limitat.

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 es pot deduir que cap nombre amb 4 dígit o més pot ser raonable (podeu demostrar-ho posant a prova els vostres coneixements sobre derivades). Per tant, podeu provar amb tots els nombres n entre 1 i 999, i amb 1000 iteracions per a cadascun d'ells en teniu suficient (veieu per què?). Aquestes iteracions consisteixen en fer $n = f(n)$, i si en algun moment arribeu al nombre inicial, llavors aquest és raonable.

Problema C6: Rajoles en fila (2).

Autora: Maria Prat

És un problema similar al del classificatori, però que cal adaptar perquè ara cada color té un pes diferent.

Repte: 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 obtenir una solució que no usi vectors i faci servir 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: Félix Moreno

Agafem tres vectors, cadascun d'ells amb els 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 dues categories. En aquest cas, aquest jugador no pot formar part de cap equip balancejat, ja que si en formés, 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 que descartem a tothom.

Pel que fa a la implementació, podem recordar qui està descartat en un vector de booleans, i podeu eliminar jugadors només esborrant l'últim element dels vectors d'habilitats.

Fixeu-vos que en el cas de només esborrar-lo en dues categories, ens podrà aparèixer un altre cop en l'altra categoria més endavant, i mirant el seu estat al vector de booleans, l'eliminem si ja ha estat descartat.

Problema C9: **Partida de dòmino.**

Autor: Salvador Roura

És un problema més llarg que difícil. Abans de començar a programar, cal planificar primer com organitzar el programa, i quines variables i funcions usar per simplificar el codi i evitar duplicacions. També cal buscar una manera raonable d'ordenar les fitxes. Al codi de Github hi trobareu alguna suggerència. També us hi mostrem una solució usant classes (els structs són una mena de classes on tot és públic per defecte) per si us interessa.