empowersoft

Concurs online de informatică
Categoria PROGRAMARE

Secţiunea 5-6 avansaţi

PROBLEMA 1 **100 puncte**

BINAR

Ionel a învăţat recent la Informatică reprezentarea numerelor în baza 2. Pentru a le aprofunda cunoştinţele, profesorul său a inventat următoarea problemă:

Dintr-un fişier text se citeşte un şir de N valori de 1 0 şi -1. Valoarea -1 are semnificaţia de terminare a unui număr, iar valorile de 0 şi 1 reprezintă cifrele în baza 2 a câte unui număr natural. Să se determine primele NR valori codificate, cu numerele de apariţii cât mai mari.

Date de intrare:
Fişierul binar.in cu următoarea structura:
- pe prima linie numerele N şi NR cu semnificaţia din enunţ.
- pe a doua linie N valori de 1 , 0 sau -1.

Date de ieşire:
Fişierul binar.out va conţine perechi distincte de numere X AP cu semnificaţia X - valoarea codificata în baza 10, AP - numărul de apariţii ale valorii X, pe fiecare linie câte o pereche despărţită printr-un spaţiu. Perechile vor fi afişate în ordinea descrescătoare a valorii AP, iar la valori egale, în ordinea descrescătoare a valorii lui X.

Restrictii şi precizări:
- 10<=N<=100000, 1<=NR<=3.
- Înaintea fiecărei valori de -1 se găseşte cel putin o valoare de 0 sau 1.
- Numerele codificate astfel sunt mai mici decat 1000 în baza 10.
- Se poate ca la stânga unui număr codificat să fie doar valori de 0.
- În şir sunt codificate cel puţin 3 valori distincte.
- Şirul de valori se incheie cu o valoare de -1.

Exemplu:

binar.in	**binar.out**	**Explicatie**
19 3 1 0 -1 1 -1 1 0 -1 1 1 -1 1 0 1 -1 1 0 1 -1	5 2 2 2 3 1	Numerele codificate sunt: 1 apare odată, 2 apare de 2 ori, 3 apare odată şi 5 apare de 2 ori. Sunt afişate primele 3 în ordinea descrescătoare a numărului de apariţii. Numerele 2 şi 5 care au acelaşi număr de apariţii se afisează în ordinea descrescătoare a valorii lor.

Timp maxim de execuţie: 1 secundă/test.
Memorie totală disponibilă: 2 MB, din care 2 MB pentru stivă
Dimensiunea maximă a sursei: 5 KB

Ploieşti, 28 mai 2016

empowersoft

Concurs online de informatică
Categoria PROGRAMARE

Secţiunea 5-6 avansaţi

PROBLEMA 2 **100 puncte**

COPACI

Pe un teren dreptunghiular de dimensiuni m şi n, din loc în loc sunt plantaţi copaci. Pentru fiecare copac se cunosc rândul şi coloana pe care este plantat, între ei fiind spaţii neplantate. Doi copaci se consideră **consecutivi** dacă mergând pe coloane, *numai de la nord către sud*, între ei sunt doar spaţii neplantate.

Cerinţă

Să se determine cea mai mare distanţă dintre doi copaci **consecutivi** şi toate perechile de copaci între care există această distanţă.

Date de intrare

Fişierul **copaci.in** conţine pe prima linie, **m**, **n** si **k** (m numărul de randuri, n numărul de coloane şi k numărul de copaci), separate prin cate un spaţiu; pe fiecare dintre următoarele **k** linii, câte o pereche de numere **x, y** (separate prin cate un spaţiu) reprezentând rândul, respectiv coloana pe care se află un copac.

Date de iesire

Fişierul **copaci.out** va conţine pe prima linie distanţa maximă dintre doi copaci consecutivi, iar pe fiecare dintre următoarele linii câte patru numere întregi **x1, y1, x2, y2** (separate prin cate un spaţiu), reprezentând coordonatele (rândul şi coloana) a doi copaci consecutivi între care distanţa este maximă.

Observaţii

- Distanţa dintre dintre doi copaci consecutivi este dată de numărul de spaţii neplantate dintre ei.
- Dacă sunt mai multe perechi de copaci între care distanţa este maximă acestea se vor afişa în ordinea crescătoare a numărului de coloană

Restricţii

- **1< m,n ≤500 şi 2 ≤ k ≤ 10000**

Exemple

copaci.in	copaci.out	Explicaţii
3 4 2 2 4 3 4	0 2 4 3 4	Sunt 2 copaci plantaţi pe coloana a 4-a şi între ei nu se află nici un spaţiu neplantat
6 7 7 1 1 1 4 2 6 3 3 4 2 5 4 6 3	8 1 1 4 2 5 4 2 6	Sunt 7 copaci. Distanţa maximă între 2 copaci consecutivi este egală cu 8. Sunt 2 perechi de copaci care se află la această distanţă. Prima pereche are copacii pe coloanele 1 respectiv 2, a doua pereche are copacii pe coloanele 4 respectiv 6.

Ploieşti, 28 mai 2016

empowersoft

Concurs online de informatică
Categoria PROGRAMARE

Secţiunea 5-6 avansaţi

Timp maxim de execuție: **0,3** secunde/test.
Memorie totală disponibilă: 2 MB, din care **2** MB pentru stivă
Dimensiunea maximă a sursei: **5** KB.

PROBLEMA 3 **100 puncte**

MUNȚI

Vrăjitorul Arpsod își dorește să își reamenajeze habitatul. În habitatul acestuia există N munți, fiecare cu o înălțime cunoscută. Fiind un tip cu un foarte dezvoltat simț estetic, el își dorește să remodeleze cei N munți astfel încât să obțină un număr maxim de munți cu aceeași înălțime. Arpsod are la îndemână o magie ce funcționează astfel: alege oricare doi munți, pe primul îl crește cu o unitate iar pe al doilea îl scade cu o unitate. Un munte poate ajunge la înălțimi negative (practic se transformă într-o groapă).
Arpsod își poate folosi magia de un număr infinit de ori.

Cerința:
Vrăjitorul vă cere să determinați numărul maxim de munți ce pot fi aduși la o înălțime egală.

Date de intrare:
Pe prima linie a fișierului ***munti.in*** se va afla numărul natural N, reprezentând numărul de munți existenți.
Pe cea de-a doua linie se vor afla N valori naturale separate prin spațiu, reprezentând înălțimea inițială a fiecărui munte.

Date de ieșire:
Fișierul ***munti.out*** va conține, pe prima și singura linie a fișierului, numărul maxim de munți ce pot fi aduși la o înălțime egală.

Restricții si precizări:
- $1 \leq N \leq 1.000.000$
- $1 \leq$ Înălțimea inițială $\leq 1.000.000.000$

Exemplu:

munti.in	munti.out	Explicație
4 2 6 2 2	4	Toți munții pot fi aduși la înălțime 3, la fiecare pas scădem o unitate din muntele de înălțime 6 și adăugăm la un munte de înălțime 2

Timp maxim de execuție: **0,5** secunde/test.
Memorie totală disponibilă: 2 MB, din care 1 MB pentru stivă
Dimensiunea maximă a sursei: **5** KB.

Ploiești, 28 mai 2016

empowersoft

Concurs online de informatică
Categoria PROGRAMARE

Secţiunea 5-6 avansaţi

PROBLEMA 4 **100 puncte**

BRAIN

Programel a fost invitat să dea o proba de angajare la cea mai mare companie de jocuri din Catania – Brain Games. Sarcina pe ca a primit-o a fost următoarea:

Scrie un program care identifică mulţimea numerelor "bine aşezate" dintr-un şir, apoi identifică cel mai mare număr care se poate obţine ca sumă de numere distincte din mulţimea determinată şi cel mai mic număr natural nenul, care **nu** se poate obţine ca sumă de numere distincte din mulţimea determinată. Un număr "bine aşezat" este un număr a cărui valoare coincide cu indicele poziţiei sale în ordinea citirii.

Date de intrare

În fişierul *brain.in* se afla pe prima linie un număr natural N, iar pe următoarea linie, un şir de N numere întregi separate prin spaţiu.

Date de iesire

a) Se va afişa pe prima linie a fişierului *brain.out* mulţimea numerelor "bine aşezate" în ordinea crescătoare a numerelor, separate printr-un spaţiu,
b) pe a doua linie cel mai mare număr care se poate obţine ca sumă de numere distincte din mulţimea determinată la punctul a.
c) pe a treia linie cel mai mic număr care **nu** se poate obţine ca sumă de numere distincte din mulţimea determinată la punctul a.

Restricţii şi precizări:

1<N<1000000;
Numerotarea poziţiilor din şir porneşte de la 1.
Se garanteaza existenţa cel puţin a unui număr şi a maxim 100000 de numere "bine aşezate" .

Se acordă 20 de puncte pentru determinarea numerelor bine aşezate, 20 puncte pentru determinarea numărului maxim şi 60 de puncte pentru determinarea numărului minim.

Exemple

brain.in	**brain.out**	**Explicaţii**
10 1 2 -5 1 3 6 7 -2 9 13	1 2 6 7 9 25 4	Numere bine aşezate sunt: 1 2 6 7 9 Numărul maxim este 25 3(1+2), 8(7+1), 10(1+2+7)……. se pot obţine ca sumă de numere din mulţimea {1, 2, 6, 7, 9} şi 4 este cel mai mic număr care nu se poate obţine ca sumă de numere distincte din mulţimea determinată
20 1 2 3 0 0 3 7 8 5 7 11 7 7 7 2 3 4 5 1 20	1 2 3 7 8 11 20 52 53	Cel mai mic număr care nu se poate obţine ca sumă de numere distincte din mulţimea {1,2,3,7,8,11,20} este 53

Timp maxim de execuţie: **0,1** secunde/test.
Memorie totală disponibilă: 4 MB, din care 2 MB pentru stivă
Dimensiunea maximă a sursei: **5** KB.

Ploieşti, 28 mai 2016