

Secțiunea 7-8 avansați

DESCRIERE SOLUȚII

PROBLEMA 1 ARTEFACTE

Autori: elevii Rareș Miroiu și Andrei Drăgan
Colegiul Național “Mihai Viteazul”, Ploiești

Soluție 100 puncte

Notând cele 3 artefacte cu A, B, C vom determina cu un algoritm de tip LEE distanțele minime de la A la toate celelalte puncte, de la B la toate celelalte puncte, de la C la toate celelalte puncte și apoi lungimea distanței minime luând în calcul toate cele 6 variante posibile

(1,1) - A - B - C - (n,n)

(1,1) - A - C - B - (n,n)

(1,1) - B - A - C - (n,n)

(1,1) - B - C - A - (n,n)

(1,1) - C - A - B - (n,n)

(1,1) - C - B - A - (n,n)

Complexitate $O(n*n)$

PROBLEMA 2 WII

Autor: elev Alexandru Dumitru
Colegiul Național “Mihai Viteazul”, Ploiești

Este nevoie de o parcurgere circulară a șirului de numere reprezentând gradele de distracție pentru a determina maximum și apoi mai este nevoie de încă o parcurgere circulară pentru a număra jocurile care satisfac condiția din enunț. De aceea, se citesc cele n numere într-un vector și se copiază încă de două ori la sfârșitul lui, obținându-se un vector cu $3*n$ elemente.

Soluția 40 puncte

Se calculează pentru fiecare joc i în parte durata sesiunii, în $ans[i]$, unde ans este vectorul soluție. Pentru aceasta se parcurg pentru jocul i elementele cu poziții cuprinse între $i+1$ și $i+2*n$ și se numără elementele care respectă condiția din enunț, numărarea oprindu-se la primul element care nu respectă această condiție. În plus, la fiecare pas se actualizează maximum dintre elementele parcurse. Dacă după această numărare se ajunge a treia oară la jocul i , adică $ans[i]=2*n+1$, sesiunea va cicla la infinit și $ans[i]$ primește valoarea -1.

Complexitatea timpului este $O(n^2)$ și complexitatea memoriei $O(3*n)$.

Soluția 200 puncte

Se parcurg cele $3*n$ numere din șir și se adaugă indicii lor într-o coadă dubla (deque), astfel încât, la pasul i primul element din coadă să rețină indicele maximum din ultima secvență prelucrată, secvență cu indici cuprinși între c și i , unde c are valoarea inițială 1. Elementele vectorului soluție ans se inițializează cu valoarea -1. La fiecare pas i , se actualizează toate elementele din vectorul soluție care au indici cuprinși între elementul curent c și lim , unde lim reprezintă indicele maximum din secvența curentă. Apoi, se actualizează indicele elementului curent c cu $lim+1$ și se extrage maximum aflat la începutul cozii, urmând a fi prelucrat următorul maxim cât timp $a[i]$ este mai mic ca jumătate din maximum anterior și coada este nevidă.

După aceea, se actualizează maximum de la începutul cozii cu elementul $a[i]$, extrăgând toate elementele mai mici sau egale ca el de la sfârșitul cozii.

Secțiunea 7-8 avansați

Algoritmul este liniar și complexitatea memoriei este $O(3*n)$.

PROBLEMA 3 PINGUINI

Autor: elevi Rareș Miroiu
Colegiul Național "Mihai Viteazul", Ploiești

Soluția 50 puncte

Pentru 40p folosim o metodă naivă, brută. Cu 4 structuri repetitive verificăm dacă fiecare cvadruplet de puncte formează un dreptunghi cu vârfurile de același rang.

Perechea (a,b,c,d) formează un dreptunghi dacă:

- Distanța între a și b = Distanța între c și d
- Distanța între a și c = Distanța între b și d
- Distanța între a și d = Distanța între b și c

Complexitate $O(N^4)$

Soluția1 100 puncte

Memorăm punctele, translatând intervalul -1000, 1000 în 0, 2000 pentru a le reține într-un tablou tridimensional (X,Y, C)

Parcurgem punctele și contorizăm câte perechi de puncte au X egal și rang egal

Complexitate $O(N^2*nrang) \approx N^2$

Soluția2 100 puncte

O soluție similară se bazează pe sortarea elementelor după rang, x și y. Se parcurge lista punctelor pentru fiecare rang în parte. În cadrul acestei parcurgerii se identifică oricare două abscise diferite x1, x2. Pentru fiecare pereche de abscise x1 și x2 se determină numărul de ordonate y care coincid ((x1,y),(x2,y)). Fie nr numărul lor. Atunci variabila sol se incrementează cu $[nr*(nr-1)/2]$.

PROBLEMA 4 PAROLA

Autor: elev Andrei Tabără
Colegiul Național "Mihai Viteazul", Ploiești

Soluția 30 puncte

Vom număra fiecare soluție posibilă, verificând dacă respectă proprietățile date.

Complexitate $O(n^2)$

Soluția 100 puncte

În loc să numărăm numărul de subșiruri bune, vom număra câte subșiruri **nu sunt bune**, să notăm numărul lor cu nr, rezultatul final fiind $(n*(n+1))/2 - nr$.

Există doar 4 tipuri de subșiruri care nu sunt bune, și anume:

1. **EEE.....EED**
2. **DDD.....DDE**
3. **EDD.....DDD**
4. **DEE.....EEE**

Secțiunea 7-8 avansați

Pentru a număra subșirurile de tipurile 1 și 2, se parcurge întâi șirul de la stânga la dreapta și se determină, în variabila k , numărul de caractere egale aflate pe poziții consecutive în șir, și de fiecare dată când se întâlnește un caracter diferit se adaugă la numărul de subșiruri care nu sunt bune valoarea k , și apoi k se reinițializează cu 1 pentru a număra următoarea secvență de caractere egale.

Pentru a număra subșirurile de tipurile 3 și 4, se parcurge din nou șirul de la dreapta la stânga și se procedează similar, dar fără a număra subșirurile care nu sunt bune formate doar din două caractere, deoarece ele au fost deja numărate în prima parcurgere.

Algoritmul este liniar, de complexitate $O(n)$.