

Secțiunea 9-10 avansați

DESCRIERE SOLUȚII

PROBLEMA 1 TEXT

Autor: prof. Raluca Costineanu
Colegiul Național “Stefan cel Mare”, Suceava**Soluție 100 de puncte**

Cerință 1

- reținem textul dat într-un tablou tridimensional cu elemente de tip char
- pentru fiecare propoziție citită reținem cuvintele ei separat pe o linie a tabloului
- determinăm lungimea maximă a unei propoziții
- afișăm pe linii separate fiecare propoziție, separând cuvintele ei prin numărul potrivit de spații, astfel încât lungimea propoziției afișate să fie egală cu lungimea maximă

Cerință 2

- salvăm toate cuvintele textului într-un vector, pe care îl ordonăm pentru a putea căuta binar fiecare subsir generat pentru primul cuvânt din text
- generarea subsirurilor unui cuvânt corespunde generării submulțimilor

PROBLEMA 2 LOL

Autor: elev Ilași Alexandru
Colegiul Național “Mihai Viteazul”, Ploiești**Soluție 100 de puncte**

Observația esențială este dată de faptul că, pentru un rol, cu siguranță va intra în echipă unul dintre cei mai buni 4 jucători disponibili (în afară de Octavian) pentru rolul respectiv la ora meciului. Astfel, pentru fiecare meci selectăm cei mai buni 4 jucători disponibili pentru fiecare rol (de exemplu, cu ajutorul unor cozi cu prioritate). Pentru fiecare meci apoi facem backtracking și alegem cea mai bună configurație. Astfel, complexitatea este $O(n \cdot 4^4 + k)$. Soluții care nu selectează în timp optim prietenii disponibili de joc pentru fiecare meci pot lua punctaje parțiale : 30/60 puncte.

PROBLEMA 3

Autor: elev Dragulin David
Colegiul Național “Mihai Viteazul”, Ploiești**Soluție 20 de puncte**

Se calculează pentru fiecare întrebare distanța maximă între campionul jucat de Octavian și orice alt campion. $O(n \cdot q)$

Soluție 30 de puncte

Calculăm soluțiile pentru cele N puncte și le reținem într-un vector, ca să răspundem la întrebări în $O(1)$. Optimizăm soluția precedentă întrucât numărul de întrebări poate fi mai mare decât numărul de campioni de pe hartă $O(n \cdot n + q)$

Soluție 100 de puncte

Observăm că toți campionii de hartă își vor folosi abilitatea pe campionii ce formează înfășurătoarea convexă formată din toate punctele ce reprezintă coordonatele campionilor. Știm că numărul de puncte de pe înfășurătoarea convexă este maxim 100, cum este precizat în enunț(H)

Secțiunea 9-10 avansați

Astfel, realizăm înfășurătoarea convexă în $O(N \log 2N)$, iar apoi calculăm pentru toți cei N campioni, care este distanța maximă față de toate punctele de pe înfășurătoare și reținem soluțiile.

Răspundem după aceste calcule la întrebări în $O(1)$. $O(N \log 2N + N * H + Q)$

PROBLEMA 4 CONSILIU

Autor: prof. Alice Georgescu
Colegiul Național "Mihai Viteazul", Ploiești

Mulțimea rangurilor poate fi asociată cu mulțimea numerelor 1,2,3..n.

1. cazul $k < n$

Problema se reduce la calculul numărului de permutări cu maxim k diferențe față de permutarea identică.

Soluția brută presupune construirea permutărilor și verificarea numărului de diferențe față de permutarea identică $O(n * n!)$.

Soluție 100 de puncte

Să considerăm cazul în care vom muta x persoane, celelate $n-x$ rămând pe pozițiile lor. Sunt combinații de n luate câte x moduri de a selecta persoanele care vor fi mutate. Vom determina numărul de moduri în care putem așeza cele x persoane selectate astfel încât să nu rămână pe poziția inițială.

Considerăm i persoana care trebuie mutată pe locul persoanei j . Sunt 2 situații posibile

1. j trece pe locul lui i și atunci rămâne să calculăm numărul de posibilități pentru $x-2$ persoane
2. j nu trece pe locul lui i și atunci problema revine la calculul numărului de posibilități pentru $x-1$ persoane, pe $x-1$ poziții dar cu $x-2$ posibilități de alegere a locului pentru fiecare persoană

Știind că sunt $x-1$ posibilități de a alege locul în care se muta persoana i avem următoarea relație:

$$nr(x) = (x-1) * (nr(x-1) + nr(x-2))$$

2. cazul $k = n$

Observația simplă pentru rezolvarea acestui caz pornește de la faptul că numerele sunt în ordine crescătoare și este suficient să facem swap între 2 poziții adiacente pentru a genera permutarea minimă lexicografic care respecta condiția ca toate elementele să nu apară în poziția inițială.