empowersoft

Concurs online de informatică
Categoria PROGRAMARE

Secţiunea 11-12

DESCRIERE SOLUŢII

PROBLEMA 1 SAH24

Autor: David Dragulin
Computer Science Imperial College London, anul I

Soluţie 100 puncte

Pentru fiecare interval [L,R], un jucător câștigă doar dacă ratingul său este egal cu cmmdc-ul valorilor din intervalul [L,R]. În plus, putem observa că ratingul câștigătorilor este minimul valorilor de rating din intervalul dat.

Pentru fiecare query vom căuta într-o structură de date (RMQ, arbori de intervale etc.) cmmdc-ul valorilor și minimul acestora. Astfel, fie M cmmdc-ul valorilor din interval și MIN minimul valorilor din interval. În funcție de aceste 2 valori vom avea 2 cazuri:

a) $M \neq MIN$: Nu există soluție pentru acest query, vom afișa 0
b) $M = MIN$: Există soluție pentru acest query. Vom afișa nrValues (numărul de elemente din interval care sunt egale cu MIN)

Algoritmul are complexitatea $O(n \cdot \log(n) \cdot \log(valMax) + Q \cdot \log(n) \cdot \log(valMax))$, unde valMax este ratingul maxim.

PROBLEMA 2 SKI

Autor: Alexandru Ilaşi
Computer Science University of Oxford, anul I

Soluţie 100 puncte

Observăm că pentru un query de tipul x y, drumul minim este următorul: x --> cel mai jos strămoș comun LCA (x, y) --> y.

Problema se reduce astfel la găsirea unui nod de întâlnire optim. Timpul de întâlnire este maximul dintre timpul parcurs de fiecare până la nodul de întâlnire.

Vom folosi:

- matricea T[k][n], care semnifica strămoșul al 2^k -lea al lui n;
- C[0][k][n], care retine costul de a cobori 2^k noduri, plecand din nodul n;
- C[1][k][n], care retine costul de a cobori 2^k noduri, ajungand in nodul n.

Odată ce am calculat LCA(x, y), urmează să verificăm care drum este mai lung (de la x la LCA sau de la y la LCA). Atunci, putem afirma faptul că nodul de întâlnire optim se află pe drumul mai lung. Pentru a găsi exact care este nodul dorit, vom efectua o căutare binară între cele 2 noduri (x sau y și LCA(x,y)), încercând să minimizăm maximul dintre drumurile parcurse de x și y.

Complexitatea algoritmului O(Q*log N* log N)

Ploiești, 27 martie 2021

empowersoft

Concurs online de informatică
Categoria PROGRAMARE

Secţiunea 11-12

PROBLEMA 3 MARFLOW

Autor: David Dragulin
Computer Science Imperial College London, anul I

Soluţie 100 puncte

Notăm val = $\sum_{i=1}^{k} a[i]$ limita superioară a soluției și q = $\prod_{i=1}^{k} a[i]$!

Știm că val este o soluție a problemei deoarece q divide val! (poate fi demonstrat folosind proprietățile numerelor prime și următoarea proprietate: $\left[\frac{n}{i}\right] + \left[\frac{m}{i}\right] \leq \left[\frac{n}{i} + \frac{m}{i}\right]$).

Plecând de la faptul că dacă q divide n!, atunci q divide (n+1)! facem observaţia că putem folosi căutarea binară pentru a găsi rezultatul minim.

Pentru fiecare număr $i \leq 10^7$, vom precalcula numărul prim maxim din descompunerea sa în factori primi, notat ciur[i], folosind ciurul lui Eratostene şi vom crea un vector cu toate numerele prime mai mici decât 10^7.

Fie vectorul cnt, unde cnt[i] = numărul de numere din a mai mari sau egale decât i

Atunci vom factoriza numitorul astfel:

```
for(int i = Max; i>=2; i--) {   // Max este valoarea maximă din șirul de intrare
  if (ciur[i] != i) cnt[ciur[i]] += cnt[i];
  cnt[i / ciur[i]] += cnt[i]; }
```

Astfel, cnt[i] = numărul de apariții a lui i în descompunerea în factori primi a numitorului.

În final, folosim căutarea binară și vectorul cnt pentru a găsi soluția minimă.

PROBLEMA 4 TRYHARD

Autor: Alexandru Ilaşi
Computer Science University of Oxford, anul I

Soluţie 100 puncte

Fie p_i probabilitatea de a trece de nivelul i. (Adică numărul P_i din enunț împărțit la 100).

Fie e_i valoarea așteptată a numărului de niveluri care trebuie jucate pentru a termina jocul, dacă se începe de la nivelul i. Prin convenție, $e_{N+1} = 0$, pentru că jocul se termină după nivelul N. Răspunsul căutat este atunci e_N.

Fiind la nivelul i, exista 2 posibilități: fie ajungem la nivelul următor, cu probabilitatea p_i, fie ne întoarcem la nivelul 1, cu probabilitatea $1 - p_i$. Atunci, putem scrie următoarele ecuații:

1. $e_1 = 1 + p_1 * e_2 + (1 - p_1) * e_1$
2. $e_2 = 1 + p_2 * e_3 + (1 - p_2) * e_1$
3. $e_3 = 1 + p_3 * e_4 + (1 - p_3) * e_1$

..

N. $e_N = 1 + p_N * e_{N+1} + (1 - p_N) * e_1$

Știind că $e_{N+1} = 0$, rămânem cu N ecuații cu N necunoscute. Rezolvând acest sistem, obținem:

$$e_N = \frac{1 + p_1 + p_1 * p_2 + p_1 * p_2 * p_3 + \cdots + p_1 * p_2 * \ldots * p_{N-1}}{p_1 * p_2 * \ldots * p_N}$$

Pentru că răspunsul trebuie dat modulo M, va trebui să înmulțim numărătorul cu inversul modular al numitorului.

Ploieşti, 27 martie 2021