Editorial RoAlgo PreOJI 2024

4-11 Martie 2024

Cuprins

1 Multumiri

Acest concurs nu ar fi putut avea loc fără următoarele persoane:

- David Curcă, Andrei Iorgulescu, Luca Mureșan, Matei Neacșu, Ștefan Vîlcescu, autorii problemelor și laureați la concursurile de informatică și membri activi ai comunității RoAlgo;
- Alex Vasiluță, fondatorul și dezvoltatorul principal al Kilonova;
- Ștefan Alecu, creatorul acestui șablon LaTeXpe care îl folosim;
- Rareș Buzdugan, Andrei Chertes, Theodor Pîrnog și ceilalți testeri ai concursului, care au dat numeroase sugestii și sfaturi utile pentru buna desfășurare a rundei;
- Andrei-Cristian Ivan, coordonatorul comisiei claselor 7-8-10;
- Comunității de informatică din România, pentru participarea la acest concurs, precum și tuturor celor care ne-au ajutat să promovăm concursul.

2 Problema egale

Autor: Luca Valentin Mureșan

Observăm că putem face în $v + 1$ operații ca toate elementele să fie egale cu v. (Putem face o operație de tip 1 urmată de v operații de tip 2, toate pe toată subsecvența (l, r)).

Soluție de 25 de puncte

Putem face toate valorile egale într-o singură operație, deci trebuie doar să verificăm dacă putem reuși acest obiectiv în 0 operații. Ca să reușim în 0 operații, trebuie să avem deja toate valorile egale cu 0. Deci, problema s-a redus la întrebări de forma ”Sunt toate valorile din (l, r) egale cu 0?”. Ca să răspundem la aceste întrebări, ne putem folosi de sume parțiale.

Soluție de 19 de puncte

Avem $n, Q \leq 1\,000$. Am vrea să aflăm în $O(n)$ răspunsul pentru o interogare. În primul rând, dacă avem vreo valoare mai mare ca v, trebuie să o setăm la 0. Acum, ca să o aducem la v, va trebui să facem înca v operații, deci în acest caz avem minim $v + 1$ operații. Deci, dacă maximul e mai mare ca v atunci răspunsul e $v + 1$.

Acum, vom face pe rând următorii pași:

1. Creștem cu 1 toate valorile de 0.
2. Creștem cu 1 toate valorile de 1.
3. Creștem cu 1 toate valorile de 2.

...

La un pas k, vrem să creștem cu 1 toate valorile de $k - 1$. Vom identifica pozițiile valorilor de $k - 1$ din șir și observăm că șirul va arăta astfel (cu X notez dacă am $k - 1$ și cu ? notez o valoare care știu că e mai mare strict decât $k - 1$):

$\underline{X}$? $\underline{X\ X}$? ? $\underline{X\ X\ X}$? ? $\underline{X}$? $\underline{X\ X}$?

Am subliniat subsecvențele pe care voi face operațiile de tipul 2. Observăm că e optim să facem pe o subsecvență maximală (nu o mai putem extinde la capete) de X uri. Așadar, vom afla aceste subsecvențe și vom crește răspunsul cu numărul de astfel de subsecvențe.

Soluție de 12 de puncte

$a_i \leq 1$, deci $a_i \in \{0, 1\}$.

Vom trata următoarele cazuri:

1. $v = 0$ Acest caz l-am tratat deja la soluția de 25 de puncte.
2. $v = 1$

Putem face deja în 2 operații, deci trebuie să verificăm dacă putem în zero sau în o operație. Ca să verificăm dacă putem în zero operații, putem doar să verificăm dacă toate valorile sunt 1 (similar cu cazul precedent). Ca să putem face o singură operație, trebuie să avem o singură subsecvență de 0. Deci, am vrea să calculăm prima și ultima poziție a lui 0 în subsecvență și să verificăm dacă între ele avem doar 0-uri. Ca să calculăm prima și ultima poziție, putem precalcula doi vectori $next0$ și $prev0$.

Soluție de 81 de puncte

Vom optimiza soluția de la subtask-ul 3 ($n, Q \leq 1\,000$). Observăm că dacă am ajuns la un moment în care am numărat deja $v + 1$ subsecvențe, ne putem opri, deoarece știm deja că avem o soluție cu $v + 1$ operații. Acum, trebuie să optimizăm cum aflăm intervalele. Putem precalcula, $next_{i,v}$ ca fiind prima apariție a valorii v în dreapta lui i. Acum, observăm că putem trece prin intervale destul de ușor în $O(maxA)$ de la un interval la altul. (unde $maxA$ e valoarea maximă din șirul a).

Soluție de 100 de puncte

Ca să ne mutăm în $O(1)$ de la un interval la altul, vom calcula $nextless_{i,v}$ ca fiind prima apariție a unei valori cel mult egală cu v în dreapta lui i. Similar, calculăm $nextgreater_{i,v}$.
Complexitate timp: $O((N + Q) \cdot VMAX)$ Complexitate timp: $O(N \cdot VMAX)$
Bonus: Găsiți soluția în $O(N + QlogN)$ timp.
Soluție oficială

3 Problema Dorel

AUTOR: MATEI NEACȘU

Subtask 1

Pentru acest subtask, un algoritm de backtracking este suficient. Un cod sursă ce rezolva subtaskul este aici.
Soluție

Subtask 2

O solutie bună pentru acest subtask este o dinamică. Vom ține in dp[i][j] numărul de aranjări posibile ale bilelor astfel încât după i cutii să luam j bile și până acum să nu avem suma mai mare ca k.
Soluție

Subtask 3 si 4

Se poate observa că atunci când $b + c = k$, nu trebuie să ținem cont de algoritmul lui Dorel, deoarece suma la final va fi mereu k așadar se va ieși din

buclă. Așadar răspunsul va fi numărul de modalități de a pune b bile în c cutii, ceea ce este știut că este $\binom{b+c-1}{b}$. Pentru subtaskul 3 se pot calcula combinările in $O(n^2)$.
Soluție

Subtask 5 si 6

Observatie rapida: daca $b + c$ nu este divizibil cu k, raspunsul este 0.
Pentru a rezolva aceste subtask-uri vom folosi stars and bars. Ca de obicei, o stea va reprezenta o bilă iar o bară finalul unei cutii. Cum avem acel +1 în algoritmul lui Dorel, o bară va reprezenta o bilă ȘI finalul unei cutii în acelasi timp rezulta ca avem b + c bile în total, b stele si c-1 bări. Avem însă restrictiile cu k care ne spun că pe pozitiile divizibile cu k trebuie obligatoriu să fie bări așadar avem deja $\frac{b+c}{k} - 1$ bări fixate deja rezulta ca răspunsul este $\binom{b+c-\frac{b+c}{k}}{c-\frac{b+c}{k}}$. Pentru subtaskul 5 se pot calcula combinările in $O(n^2)$.
Soluție

4 Problema Pofta

AUTOR: DAVID-IOAN CURCĂ

Soluție de 40 de puncte

Pentru cel puțin 40 de puncte, putem calcula pentru fiecare pereche de persoane (i, j) compatibilitatea lor (desigur, dacă este respectată condiția $dist(i, j) \le depmax_i$). În timpul concursului au fost obținute și punctaje mai mari de 50 de puncte parcurgând până la ultima poziție candidată. Complexitate: $O(N^2)$

Soluție completă

Ne amintim că $comp(i, j) = |d_i - d_j| \cdot (-coef) + f_j$. Considerăm cele două cazuri pentru $|d_i - d_j|$.

Cazul 1: $d_i > d_j \Leftrightarrow i > j$

$comp(i, j) = (d_i - d_j) \cdot (-coef) + f_j$
$comp(i, j) = d_i \cdot (-coef) - d_j \cdot (-coef) + f_j$
$comp(i, j) = d_j \cdot coef + f_j + d_i \cdot (-coef)$

Pentru a afla j optim pentru un i fixat, observăm că $d_i \cdot (-coef)$ este constant, deci vrem să găsim $d_j \cdot coef + f_j$ maxim. Dar $d_i - d_j \leq depmax_i \Rightarrow d_j \leq d_i - \text{depmax}_i$. Secvența formată din $d_i - depmax_i$ este un șir crescător, ceea ce înseamnă că putem folosi o structură de date precum deque pentru a afla elementul unde $d_j \cdot coef + f_j$ este maxim, parcurgând de la stânga la dreapta și eliminând elementele din coadă care nu respectă $d_j \leq d_i - depmax_i$.

Cazul 2: $d_i < d_j \Leftrightarrow i < j$

$comp(i, j) = (d_j - d_i) \cdot (-coef) + f_j$
$comp(i, j) = d_j \cdot (-coef) - d_i \cdot (-coef) + f_j$
$comp(i, j) = d_j \cdot (-coef) + f_j + d_i \cdot coef$
În acest caz, $d_i \cdot (-coef)$ este constant, acum trebuie găsit $d_j \cdot (-coef) + f_j$ maxim. De data aceasta, șirul va fi parcurs de la dreapta la stânga.
Complexitate: $O(N)$