Editorial RoAlgo Contest 9

4 Februarie 2024

Cuprins

1 Multumiri

Acest concurs a fost primul concurs cu propunători internaționali, fiind un mirror a unei alte competiții organizate anterior. Acest concurs nu ar fi putut avea loc fără următoarele persoane:

- Ștefan-Cosmin Dăscălescu, Kyle Liang și Rahul Verma, autorii problemelor și membri diverselor comunități internaționale de informatică;
- Alex Vasiluță, fondatorul și dezvoltatorul principal al Kilonova;
- Ștefan Alecu, creatorul acestui șablon LaTeXpe care îl folosim;
- Ethan Lee, Thomas Liu și Julian Wu, testerii și participanții experimentați ai concursului, care au dat numeroase sugestii și sfaturi utile pentru buna desfășurare a rundei;
- Ștefan-Cosmin Dăscălescu, coordonatorul rundei;
- Comunității RoAlgo, pentru participarea la acest concurs.

2 Square or Prime Digit?

Autor: Stefan Dascalescu

2.1 Soluția oficială

Deoarece numărul dat are cel mult 100 de cifre, va trebui să-l procesăm ca un șir de caractere, deoarece tipurile de date int și long long din C/C++ pot păstra numere de cel mult 10, respectiv 19 cifre.
Astfel, vom parcurge șirul și pentru fiecare cifră, vom verifica dacă este un pătrat perfect $(0, 1, 4, 9)$ sau un număr prim $(2, 3, 5, 7)$.
Vom ține un contor pentru fiecare din aceste valori și la final, vom afișa mesajul potrivit în funcție de numărul de cifre pătrate și numărul de cifre prime.

2.1.1 Cod sursă

Soluție de 100

3 Greedy Gary and GCDs

Autor: Rahul Verma

3.1 Soluția oficială

În primul rând, $cmmdc(a, b) \cdot cmmmc(a, b) = a \cdot b$. Putem rescrie această proprietate în diverse moduri, pentru a putea afla una din valori în funcție de celelalte două.
În al doilea rând, dacă unul din numere nu este divizibil cu celălalt, răspunsul este 0 deoarece nu vom avea vreo soluție număr întreg.
Apoi, avem trei cazuri pentru ultimul număr - fie este produsul lor, fie cmmdc-ul lor, fie cmmmc-ul lor. Dacă este produsul lor, înseamnă că celelalte două sunt cmmdc și cmmmc, le vom putea înmulți. Altfel, vom împărți numărul mai mare la numărul mai mic.
Va trebui să avem grijă să nu numărăm de mai multe ori aceeași soluție (dacă cele două valori sunt egale, trebuie să afișăm 1, nu 2).
Astfel, pe scurt, soluția finală este următoarea:

- 0: $\max(v_1, v_2) \mod \min(v_1, v_2) \neq 0$
- 1: $\max(v_1, v_2) \mod \min(v_1, v_2) = 0$ și ($v_1 = 1$ sau $v_2 = 1$)
- 2: $\max(v_1, v_2) \mod \min(v_1, v_2) = 0$

3.1.1 Cod sursă

Soluție de 100

4 Bit Showdown

Autor: Stefan Dascalescu

4.1 Soluția oficială

Pentru această problemă, observația principală pe care trebuie s-o facem este aceea că putem trata fiecare bit independent de ce se întâmplă cu ceilalți biți. Astfel, pentru fiecare bit de la 0 la 19, vom putea folosi sume parțiale pentru a număra câte dintre valorile șirului au acel bit setat (egal cu 1), iar pentru a răspunde la fiecare query, vom avea grijă să ne folosim de proprietățile operațiilor pe biți descrise în enunț.
Voi nota cu cnt_1 numărul de 1 din intervalul dat, respectiv cu cnt_0 numărul de 0 din acel interval. De asemenea, voi nota cu len lungimea intervalului. Se va putea afla răspunsul ușor folosind aceste formule pentru fiecare bit.

- AND: pentru fiecare bit i, vom adăuga $2^i \cdot \frac{cnt_1 \cdot (cnt_1 - 1)}{2}$
- OR: pentru fiecare bit i, vom adăuga $2^i \cdot (len^2 - \frac{cnt_0 \cdot (cnt_0 - 1)}{2})$
- XOR: pentru fiecare bit i, vom adăuga $2^i \cdot cnt_0 \cdot cnt_1$

4.1.1 Cod sursă

Soluție de 100

5 Larry the Lizard

Autor: Rahul Verma

5.1 Soluția oficială

Mai întâi, vom ordona descrescător testele în funcție de diferența dintre punctajul după ce înveți și punctajul inițial. Apoi, creăm sume parțiale pentru diferențele menționate mai sus.
Apoi, pentru fiecare valoare pe care o scoatem, vom putea căuta binar răspunsul cerut de problemă. La un pas al căutării binare, vom vrea să vedem dacă putem ajunge la Q puncte studiind acele teste, dar trebuie să avem grijă să evităm numărarea acelui test pe care l-am scos de două ori, acest lucru tratându-se fără mari probleme în interiorul căutării binare.
În final, complexitatea soluției va fi $O(N \log N)$

5.1.1 Cod sursă

Soluție de 100

6 City Skylines

Autor: Kyle Liang

6.1 Soluția oficială

O primă observație este aceea că dintre toate nodurile inițiale, adăugarea unui nod nou va afecta doar un nod din cele inițiale. Astfel, putem trata fiecare nod în mod independent.
O altă observație importantă este aceea că putem să tratăm diversele cazuri care pot apărea pentru a echilibra fiecare nod, concentrându-ne pe valoarea nodului (v) și suma nodurilor adiacente (u). Tot ce ne rămâne de făcut este să tratăm aceste cazuri și să implementăm soluția, în funcție de aceste două valori menționate.

6.1.1 Cod sursă

Soluție de 100

7 Lucky Sudoku

Autor: Kyle Liang, Stefan Dascalescu

7.1 Soluția oficială

Pentru a rezolva această problemă, va trebui să avem grijă la cum păstrăm punctele, precum și la punctele care ne-ar putea face să nu putem folosi diverse linii sau coloane. Mai întâi, putem afla ușor ce linii nu vor fi considerate la răspuns (cele care au cel puțin un 8), iar mai apoi la o primă parcurgere a punctelor date, vom ține și coloanele care au un 8 în ele.
Apoi, pentru celelalte puncte care nu sunt 8 iar coloanele lor nu au fost scoase din cauza valorilor de 8, va trebui să mai ștergem din valorile pe care le putem pune pe liniile respective.
Până acum am tratat cazurile legate de aceeași linie și aceeași coloană. Mai trebuie să tratăm cazurile când trebuie să scoatem puncte deoarece există un 8 în zona de $3 \cdot 3$ corespunzătoare acelui pătrat. Pentru a face asta, va trebui să considerăm doar zonele în care există cel puțin un pătrat umplut, verificând pentru fiecare punct dacă a fost scos din calcule anterior, iar dacă nu a fost scos și avem un 8 în acea zonă, îl vom scoate din calcule.
Astfel, complexitatea finală a soluției va fi $O(M \log M)$

7.1.1 Cod sursă

Soluție de 100

8 Lark and Nightingale

Autor: Kyle Liang

8.1 Soluția oficială

Pentru a rezolva această problemă, va trebui să calculăm sumele parțiale pentru A și pentru B, având grijă să rotim matricea B la 45 de grade pentru a putea transforma sumele pe diagonală în sume pe linii.
Acum, soluția va fi identică pentru ambele tipuri de valori. Pentru fiecare pătrat, putem căuta binar cât putem extinde construcția de tip pătrat sau romb, verificând cu sumele parțiale dacă acea arie este acoperită complet de A sau B, după caz.
Pentru mai multe detalii de implementare, puteți consulta soluția oficială.

8.1.1 Cod sursă

Soluție de 100