RoAlgo Contest 1 - Editorial

David-Ioan Curcă, Ștefan-Cosmin Dăscălescu, Rareș Drăgușanu
Luca Valentin Mureșan, Vlad Mihai Tutunaru

January 2023

1 Mulțumiri

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

- David-Ioan Curcă, Rareș Drăgușanu, Luca Valentin Mureșan, Vlad Mihai Tutunaru, autorii problemelor, laureați la concursurile de informatică de gimnaziu și membri ai lotului național de juniori.
- Alex Vasiluță, fondatorul și dezvoltatorul principal al Kilonova
- Tiberiu-Ștefan Cozma, testerul concursului și cel care a dat numeroase sugestii și sfaturi utile pentru buna desfășurare a rundei.
- Ștefan-Cosmin Dăscălescu, coordonatorul rundei și autorul uneia dintre probleme.
- Alexandru Peticaru, pentru citatul inspirațional, care ne-a încurajat să terminăm acest concurs.
- Comunității RoAlgo, pentru participarea la acest concurs.

2 Problema Expansion

Autor: Ștefan-Cosmin Dăscălescu

2.1 Soluții parțiale

Pentru subtaskul 1, putem afișa 0 deoarece singura zonă în care am putea avea pătrate libere e pe marginea matricii și după o primă extindere, oricum s-ar așeza R-urile, nu ar mai fi conectate.

Pentru subtaskul 2, putem încerca fiecare lungime în parte pentru a verifica care este răspunsul final, folosind algoritmul lui Lee, extinzând zidurile cu cantitatea corespunzătoare pasului ales. Complexitatea totală va deveni $O(n * m * max(n, m))$.

2.2 Soluția completă

Putem optimiza subtaskul 2 folosind o căutare binară pe răspuns, deoarece dacă putem extinde zidurile cu x fără să afectăm conexiunile dintre R-uri, putem extinde și cu mai puțin de x și R-urile tot ar fi conectate unele cu altele. Complexitatea totală va deveni $O(n * m * \log n)$.

Cod sursă: Solutie

3 Problema Gadfadăr3

AUTOR: LUCA VALENTIN MUREȘAN

O observație inițială este că putem considera că $a_i \leq b_i$.

Vom fixa valoarea x și vom încerca să facem fiecare element din șirul c cât mai aproape de elemetul x.

Știm mediana șirului c. Așadar, când vrem să căutăm valoarea x care va minimza $f(c)$ dintre toate șirurile c, vom parcurge doar elementele care se află în șirul a sau în șirul b.

Această soluție ne aduce la complexitatea temporară $O(N^2)$.

Pentru soluția de 100 de puncte, vom fixa valoarea i și observăm că pentru poziția i, putem distinge următoarele 4 cazuri:

Notăm $s = (a_i + b_i)\ /\ 2$.

1. Dacă $x < a_i \leq b_i$:

Pe poziția i, răspunsul va crește cu $a_i - x$.

2. Dacă $a_i \leq x < s$:

Pe poziția i, răspunsul va crește cu $x - a_i$.

3. Dacă $s \leq x < b_i$:

Pe poziția i, răspunsul va crește cu $b_i - x$.

4. Dacă $b_i \leq x$:

Pe poziția i, răspunsul va crește cu $x - b_i$.

Pentru a rezolva problema, vom normaliza valorile și vom utiliza de două ori Jmenul lui Mars sau vom folosi baleiere.

tibinyte - baleiere

LucaLucaM - Mars

Complexitate temporală $O(N \log N)$.

4 Problema Abmnk

Autor: Rareș Drăgușanu

4.1 Cerinta 1

Se observa ca nu avem nevoie sa stocam numerele din a, suma lor ramanand aceeasi, indiferent de ordinea din v.

Putem face un vector "puteri" in care precalculam toate puterile numerelor din b. Apoi, pentru fiecare numar gasim cea mai apropiata putere. Aceasta solutie obtine 11 puncte.

Putem eficientiza solutia anterioara, sortand puterile. Astfel, pentru fiecare numar putem cauta binar cea mai apropiata putere. Aceasta solutie obtine 29 de puncte

4.2 Cerinta 2

O soluție de tip brute-force obține 19 puncte.

Pentru 71 de puncte, observăm faptul că valoarea maximă a lui m este cel mult 9 face numărul de puteri distincte să fie foarte mic, ceea ce ne permite să ținem doar un bit de memorie care ne zice poziția puterii în șir, urmând ca mai apoi să efectuăm parcurgerea din k în k, optimizând memoria prin acest procedeu.

Cod sursă: Solutie

5 Problema Echipe

Autor: David-Ioan Curcă

Raspunsul final al problemei este produsul numerelor de moduri pentru fiecare sub-echipă. Este clar că șirul p este o permutare. Aceasta permutare trebuie să fie descompusă in cicli. Nu contează ce elemente formează acel ciclu, ci doar numărul de jucători din ciclul respectiv.

Brutul la această problemă obține 12 puncte.

Pentru incă 40 de puncte se poate folosi o abordare bazată pe metoda programării dinamice. Definim

$$dp_{i,j} = \text{numărul de moduri de a face o sub-echipă cu } i \text{ elevi și } j \text{ roluri}$$

$$dp_{i,j} = dp_{i-1,j-1} \cdot j + dp_{i-1,j} \cdot j$$

Această abordare nu obține 100 de puncte deoarece nu avem destulă memorie.

Soluția oficială folosește Principiul Includerii Și Excluderii. Numărul total de moduri de a da roluri echipei indiferent de corectitudine este k^n, unde n aici reprezintă lungimea ciclului, nu a șirului. Daca fixam ca j roluri să nu apară în echipă o să fie $(k-j)^n$ moduri de a da roluri. De asemenea, avem C_k^j moduri de a alege acele j roluri excluse. Deci formula finală devine

$$k^n - \sum_{j=1}^{k} (-1)^{j+1} \cdot C_k^j \cdot (k-j)^n$$

Calculele pentru combinari vor fi facute folosind invers modular și precalculând inversele factorialelor. Vă reamintim că:

$$\frac{1}{i!} \equiv (i+1) \cdot \frac{1}{(i+1)!} \pmod{10^9+7}$$

Complexitatea finală este $O(\sqrt{n} \cdot k \cdot \log k)$. De menționat faptul că soluția oficială folosește memoizare pentru a nu calcula de mai multe răspunsul pentru aceeași lungime.

Cod sursă: Solutie

6 Problema Cdcq

Autor: Luca Valentin Mureșan

După niște căutări aprigi, observăm că *haiestra* unui număr este de fapt cifra de control. Deci dacă pentru un query de forma (x, y), avem cel puțin un 0 de la x la y atunci produsul este 0, deci și cifra de control este 0. Dacă nu, fie $cnt_{j,i}$ de câte ori apare un număr care dă restul j la împărțirea cu 9. Acum produsul este

$$\prod_{j=0}^{8} j^{cnt_{j,y} - cnt_{j,x-1}}$$

$(mod\ 9)$, sau 9 dacă produsul este chiar divizibil cu 9.

De asemenea, se încadrau în timp și surse cu arbori de intervale.

LucaLucaM - Sursa Oficială

tibinyte - Arbore de intervale