RoAlgo Contest 4 - Editorial

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

August 2023

1 Mulțumiri

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

- David-Ioan Curcă, 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, Matei Neacșu, Andrei Chertes, Traian Danciu, testerii concursului și cei care au dat numeroase sugestii și sfaturi utile pentru buna desfășurare a rundei.
- Simion Lavrente, pentru desenele de la leximin și ast.
- Ștefan-Cosmin Dăscălescu, coordonatorul rundei.
- Alexandru Peticaru, pentru citatul inspirațional, care ne-a încurajat să terminăm acest concurs.
- Comunității RoAlgo, pentru participarea la acest concurs.

2 Problema Poseidon

Autor: Luca Valentin Mureșan

2.1 Soluții parțiale

Pentru subtaskul 1, putem fixa lungimea bărci, după aceea să verificăm în timp liniar dacă există un șir cu acea lungime.

Pentru subtaskul 2, observăm că putem căuta binar lungimea bărci și să verificăm în timp liniar dacă există un subșir cu acea lungime.

Pentru subtaskul 3, observăm că putem verifica în timp logaritmic dacă există un subșir cu lungimea fixată. Vom căuta binar prima poziție pentru care vom avea suficiente caractere de fiecare tip.

2.2 Soluția completă

Observăm că putem optimiza verificarea pentru o lungime fixată. Vom precacula pentru fiecare din cele 3 caractere ale unei bărci, $first_i =$ prima poziție pentru care avem i caractere de același tip.

Complexitate temporală: $O(N + Q\ logN)$

Cod sursă: Solutie

3 Problema Leximin

AUTOR: LUCA VALENTIN MUREȘAN

Observăm că ne-ar ajuta foarte mult să calculăm nodurile din care putem ajunge în nodul K. Așadar, vom face un dfs/bfs pe graful transpus. Acum, vom porni un dfs din nodul cu valoarea minimă din care putem ajunge în nodul K. Acum, ne vom duce greedy în cel mai mic nod prin care înca nu am trecut și din care putem ajunge în nodul K.

Complexitate temporală $O(N + M)$.

Cod sursă: Solutie

4 Problema Ast

AUTOR: DAVID-IOAN CURCĂ

Vom face un dfs din nodul rădăcină și vom calcula pentru fiecare nod, valoarea obținută dacă considerăm doar subarborele acelui nod.

Complexitate temporală $O(N)$

Cod sursă: Solutie

5 Problema Cenzura

AUTOR: LUCA VALENTIN MUREȘAN

Ne vom calcula pentru fiecare prefix și pentru fiecare sufix, care e valoarea maximă pe acel prefix / sufix și de câte ori apare. Pentru fiecare interogare, vom combina prefixul de lungime $l-1$ cu sufixul de lungime $r+1$.

Complexitate temporală $O(N+Q)$

Cod sursă: Solutie

6 Problema chmax

Autor: Luca Valentin Mureșan

Ne ajută foarte mult faptul că $l \leq p \leq r$. Așadar, putem precalcula pentru fiecare poziție i valorile st_i și dr_i cu următoarea semnificație: $st_i =$ cea mai mare poziție astfel încat $st_i < i$ și $a_{st_i} > a_i$. Analog, definim și dr_i.

Aceste precalculări le putem face folosind stive.

Acum, pentru fiecare interogare trebuie doar să verificăm dacă $st_p < l$ și $dr_p > r$.

Complexitate temporală: $O(N+Q)$

Cod sursă: Solutie

7 Problema trap

Autori: David-Ioan Curcă, Luca Valentin Mureșan

Este bine știut faptul că $a \otimes b + 2(a \ \& \ b) = a + b$. De aici reiese că $a \otimes b + (a \ \& \ b) = a + b \iff a \ \& \ b = 0$. Ne vom folosi de următoarea dinamică pe cifre: $dp_{i,LE/GEQ,LE/GEQ,LEQ/GR,LEQ/GR}$ unde LE vine de la less, GR de la greater, LEQ de la less or equal și GEQ de la greater or equal. Semnificația este următoarea: $a \geq l$ sau $a < l$, $b \geq l$ sau $b < l$, $a \leq r$ sau $a > r$ și $b \leq r$ sau $b > r$. Vom calcula dinamica pe sufix (de la dreapta la stânga). Răspunsul se află în $dp_{1,GEQ,GEQ,LEQ,LEQ}$ Acest tip de dinamică este cunoscut sub numele de digit dp. link.

Complexitate temporală: $O(N)$

Cod sursă: Solutie

Matei Neacșu: Solutie alternativa

8 Problema Slayer2

Autor: Vlad Tutunaru

Să presupun ca Ian ia pietrele $x_1, x_2 \ldots x_k$ și Robi pietrele $y_1, y_2 \ldots y_{n-k}$. În ordinea aceasta.

Observăm că pentru o pereche de secvențe x, y există doar o permutare p care corespunde perechii.

Notăm cu S suma elementelor din șirul a.

Deci trebuie să numarăm numărul de perechi x, y cu aceeași sumă a elementelor. Trebuie să calculăm numărul secvențelor de sumă $\frac{S}{2}$ de lungime K. O soluție este folosirea programării dinamice.

Complexitate temporală: $O(N^2\dot{S})$.

Cod sursă: Solutie

9 Problema Gadfadăr5

Autor: Luca Valentin Mureșan

Prima observație:

Dacă reușim să calculăm cnt_d = câte subsecvențe au ambele capete (i, j) divizibile cu d. Dacă am reușit să calculăm cnt_d pentru fiecare d de la 1 la N, putem face un ciur invers pentru a calcula $answer_d$ = câte subsecvențe au $gcd(i, j) = d$.

A doua observație:

În continuare, ne vom putea folosi de o structură de date (de exemplu un RMQ) pentru a răspunde rapid la întrebări de forma:

"Este subsecvența $[l, r]$ corect parantezată?"

Pentru a răspunde la astfel de întrebări, ne vom folosi de două proprietăți necesare și suficiente pentru ca un șir să fie corect parantezat:

1. Numărul de paranteze deschide trebuie să fie egal cu numărul de paranteze închise.

2. Pentru orice prefix, numărul de paranteze închise nu depășește numărul de paranteze deschise.

Soluție:

Pentru ușurință, vom nota cu $delta_i$ = diferența între numărul de paranteze deschise și numărul de paranteze închise dintre primele i caractere (putem calcula ușor).

Pentru a calucla cnt_d, vom calcula pentru fiecare i divizibil cu d, dp_i = numărul de șiruri corect parantezate care încep pe poziția i. Avem următoarele două cazuri:

1. Începem un șir nou

Vrem să vedem numărul de valori j pentru care j este divizibil cu d și subsecvența $[i, j]$ este corect parantezată. Vom căuta binar cea mai din dreapta poziție k (k nu neapărat divizibil cu d) și vom verifica în subsecvența $[i, k]$ numărul de valori j divizibile cu d pentru care $delta_j = delta_{i-1}$. Vom aduna la dp_i numărul de astfel de j-uri.

2. Continuăm o subsecvență

Vrem să vedem prima poziție j pentru care subsecvența $[i, j-1]$ este corect parantezată și j este divizibil cu d. Vom adăuga dp_j la dp_i.

Acum, este simplu de observat că $cnt_d = dp_d + dp_{2d} + dp_{3d} + \dots$.

A treia observație:

O mică optimizare este să considerăm doar valorile d impare, deoarece $gcd(i, j) = d \iff gcd(i, j-i) = d$, dar lungimea unui șir corect parantezat este pară ($j - i + 1$ divizibil cu 2). Dacă d ar fi par ar rezulta că $j - i$ ar fi par, deci $j - i + 1$ ar fi impar, ceea ce e imposibil. Totuși, această optimizare nu era necesară pentru obținerea punctajului maxim.

Complexitate temporală: $O(N \ \log^2 N)$

Cod sursă: Solutie