Editorial RoAlgo PreOJI 2024

4-11 Martie 2024

Cuprins

1 Multumiri

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

- Luca Valentin Mureșan, Lucian Andrei Badea, Ștefan Vîlcescu și Traian Mihai Danciu, 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 și ceilalti testeri ai concursului, care au dat numeroase sugestii și sfaturi utile pentru buna desfășurare a rundei;
- Ștefan-Cosmin Dăscălescu, coordonatorul comisiei claselor 5-6-9;
- Comunității de informatică din România, pentru participarea la acest concurs, precum și tuturor celor care ne-au ajutat să promovăm concursul.

2 Problema sirpal

Autor: Luca Valentin Mureșan

Soluție de 12 de puncte

Pentru $n = 3$, e clar că pozițiile 1 și 3 răspunsul e 0. Pentru poziția 2, trebuie să verificăm dacă $a_1 = a_3$.

Soluție de 56 de puncte

Observăm că dacă avem un șir $x_1 < x_2 < \ldots, x_k$ care conține poziția i și $x_1 \neq i \neq x_k$, atunci putem alege șirul (x_1, i, x_k). Deci, avem un șir palindromic care îl conține pe i dacă și numai dacă avem un șir palindromic de lungime 3 care îl conține pe i. Ca să verificăm acest lucru, putem doar să verificăm dacă există două numere egale, unul în stâng lui i și unul în dreapta lui i. Pentru $n \leq 300$, putem să verificăm toate perechiile. Pentru $n \leq 2\,000$, putem să facem un vector de frecvență pentru numerele din stânga lui i, iar pentru numerele din dreapta lui i, putem doar să verificăm dacă apare cel puțin o dată în vectorul de frecvență.

Soluție de 100 de puncte

Acum, trebuie să optimizăm cum verificăm pentru fiecare i. Există mai multe metode, una dintre este următoarea:
Pentru fiecare valoare, observăm că e importantă doar prima și ultima apariție a acesteia (dacă am folosi una care se află între, am putea defapt doar să alegem cele două de pe margine). Acum, orice valoare cuprinsă strict între cele două poziții, ne va da răspunsul 1. Deci, putem să folosim Jmenul lui Mars ca să adăugam 1 pe acel intreval. În final, pentru fiecare valoare vom afișa 0 sau 1 dacă am adăugat sau nu ceva pe acea poziție.
Complexitate timp și memorie: $O(N + VMAX)$
Soluție oficială

3 Problema Sircost

Autor: Ștefan Vîlcescu

Soluția de 13 puncte

Deoarece $a_i = a_{i+1}$, atunci pentru fiecare query, $a_{dr-1} = a_{dr}$, deci rezultatul va fi mereu 0

Solutia de 26 puncte

Fie $nrplus_i$=numărul de indici $j \leq i$ astfel încât $a_j < a_{j+1}$ Fie $nrminus_i$=numărul de indici $j \leq i$ astfel încât $a_j > a_{j+1}$ Fie sp_i=răspunsul pentru queryul $[1, i]$
Dacă $st = 1$, vom afișa sp_{dr}. Altfel, în subsecvența $[st, dr]$, $sp_{dr} - sp_{st}$ va rezulta rezultatul query-ului, doar că numerele adunate sau scăzute vor fi $st + 1, st + 2, st + 3, \ldots, dr$, nu $1, 2, 3, \ldots, dr - st$. De exemplu, se dă șirul

9 10 4 5 1 133 12

Pentru queryul $[3, 6]$, dacă facem $sp_{dr} - sp_{st}$, rezultatul nostru va fi $-2 + 3 - 4 = -3$, însă rezultatul corect este $-1 + 2 - 3 = -2$. Cum rezolvăm aceasta? Dacă avem x numere negative în rezultatul nostru, acestea x trebuie

adunate cu $(l-1)$ pentru a rezulta că noi vom scădea numerele corecte. De asemenea, dacă avem y numere pozitive în rezultatul nostru, acestea $(y-1)$ trebuie scăzute cu $(l-1)$ pentru a rezulta că noi vom aduna numerele corecte.
Deoarece avem șirurile nr_{plus} și nr_{minus}, rezultatul nostru va fi $sp_{dr} - sp_{st} - (l-1)(nr_{plusdr} - nr_{plusst}) + (l-1)(nr_{minusdr} - nr_{minusst})$.
Deoarece știm că elementele șirului sunt distincte iar vecinii unui element sunt distincți, nu există pericolul de a ne transforma costul șirului în 0.

Solutia de 24

Deoarece $1 \leq N, Q \leq 2000$, queryurile se pot face brut, complexitatea fiind $O(N * Q)$

Solutia de 100 de puncte

Singura diferență între soluția completă și soluția de 26 de puncte este cum rezolvăm când avem elemente vecine egale.
Răspunsul este foarte simplu:
Fie $ultim_i$ = ultimul indice $j \leq i$ astfel încât $a_j = a_{j+1}$.
Acum avem două cazuri:
Dacă $ultim_{dr} \leq st$, se va aplica formula de la soluția de 26 de puncte.
Altfel, se va aplica tot formula de la soluția de 26 de puncte, doar că de data aceasta vom înlocui st cu $ultim_{dr}$, mai putin unde inmultim cu $(st-1)$, deoarece numerele pe care le adunam/scadem inca sunt 1, 2, 3, ..., dr-s
Soluția oficială

4 Problema factoria

Autor: Traian Mihai Danciu

Solutie de 10 de puncte

Mai întâi, putem observa că ne putem baza pe descompunerea în factori primi.

Teorema 4.0.1. *Un număr a este divizibil cu alt număr b dacă și numai dacă pentru fiecare p număr prim, dacă p este factor prim al lui b, exponentul lui p în descompunerea în factori primi a lui b este mai mic sau egal cu exponentul lui p în descompunerea în factori primi a lui a.*

Observație. Daca p este un număr prim și n este un număr natural, astfel încât n nu este divizibil cu p, atunci exponentul lui p în descompunerea în factori primi a lui n este 0.

Mai întâi, vom afla pentru fiecare număr prim (care divide factorialul lui k), exponentul lui în descopmunerea în factori primi a factorialului lui k. Vom face acest lucru astfel: pentru fiecare număr de la 1 la k, îl vom descompune în factori primi, și pentru fiecare factor prim, adaugăm la un vector de frecvență, pe care îl vom nota $frecv$[], exponentul lui. Vectorul de frecvență va fi același pentru fiecare număr prim.

Apoi, pentru a număra toate subsecvențele care au produsul divizibil cu factorialul lui k, vom itera prin toate subsecvențele. Mai întâi, vom fixa capătul stânga l, de la 1 la n, iar apoi vom fixa capătul dreapta r de la l la n.
După ce am fixat capetele secvenței, vom itera prin toate elementele din secvență, le vom descompune în factori primi, și vom adăuga la un alt vector de frecvență, $curent[]$, exponentul lui.
La final vom verifica dacă pentru fiecare număr prim p,
$curent[p] \geq frecv[p]$. Dacă acest lucru se întâmplă pentru fiecare număr p, atunci produsul elementelor din subsecvența curentă este divizibil cu factorialul lui k, și adăugăm 1 la un contor, care va reprezenta răspunsul final.
După ce verificăm o subsecvență, vom reseta vectorul $curent[]$.

Soluție de 30 de puncte

Ne vom baza pe soluția anterioară, doar că în loc să verificăm fiecare subsecvență după ce fixăm și capătul stânga și capătul dreapta, vom face astfel: fixăm doar capătul stânga.
Apoi, iterăm cu capătul dreapta r, și vom descompune numărul de pe poziția r în factori primi și vom adăuga exponentul fiecărui factor prim la $curent[]$.
Astfel, noi am adăugat fiecare număr de la l la r.
Apoi, verificăm subsecvența exact ca la soluția anterioară.
După ce iterăm cu capătul dreapta, vom reseta vectorul $curent[]$.

Soluție de 65 de puncte

Din nou, ne vom baza pe soluția anterioară, doar că în loc să verificăm fiecare subsecvență fixând capetele stânga și dreapta, ne vom folosi de o metodă care

ne ajută în multe probleme în care trebuie să numărăm câte secvențe ale unui șir respectă o proprietate: metoda two pointers.
Mai întâi, vom itera cu capătul stânga de la 1 la n. Apoi, capătul dreapta r nu va fi o variabilă care se va folosi doar pentru acest capăt stânga, ci va fi o variabilă folosită pe tot parcursul algoritmului, care mai întâi va avea valoarea 0. Ea va reprezenta ultima valoare adăugată la subsecvența curentă.
Apoi, după ce am iterat cu capătul stânga l, pentru fiecare iterație vom face astfel: cât timp $r < n$, adică cât timp mai avem elemente de adăugat, și cât timp subsecvența curentă nu respectă proprietatea, vom incrementa r și vom descompune în factori primi elementul de pe poziția r și vom adăuga exponentul fiecărui factor prim la $curent[]$.
Apoi, dacă subsecvența curentă respectă proprietatea, ea ar fi respectat-o pentru orice capăt stânga de la r la n. Rezultă că vom avea $n - r + 1$ secvențe cu capătul stânga în l.
Apoi, vom descompune în factori primi numărul de pe poziția l și vom scădea din $curent[]$ exponentul fiecărui factor prim.

Soluție de 100 de puncte

Și aici ne bazăm pe soluția anterioară. Observăm mai întâi că noi facem foarte multe descompuneri în factori primi, iar cum valorile sunt până în 10^6, noi am putea precalcula, folosind ciurul lui Eratostene, pentru fiecare număr x, $ciur[x]$ = cel mai mic divizor prim al lui x. Apoi, vom descompune în factori primi astfel: cât timp numărul este mai mare ca 1, divizorul curent va fi valoarea lui în $ciur[]$.
Apoi, ca să verificăm dacă o secvență, vom menține un contor, care ne va spune pentru câte numere i de la 1 la k, $curent[i] \geq frecv[i]$. La început, contorul are valoarea numărul de numere care nu sunt prime de la 1 la k.

Astfel, o secvență are proprietatea cerută dacă contorul are valoarea k.

Soluție parțială de 10 puncte

Soluție parțială de 30 de puncte

Soluție parțială de 65 de puncte

Soluție oficială