

IOT 2021-22 Runda 3 - Soluții

Comisia IOT

Ianuarie 2022

1 Problema Bitwise Party

AUTOR: STEFAN DASCALESCU

1.1 Descrierea soluției

Dificultate: Medie

Pentru primul subtask, trebuie ținut minte numărul de 1 și numărul de 2 și răspunsul e maximul dintre cele două valori. Trebuie avut grijă la paritatea răspunsului (XOR-ul dintre un număr par de elemente egale este 0).

Pentru cel de-al doilea subtask, putem rezolva problema folosind o soluție bazată pe metoda programării dinamice. Astfel, $dp[i][j][k]$ e egal cu 1 dacă din primele i numere, putem alege numere astfel încât AND-ul să fie egal cu j și XOR-ul să fie egal cu k . De asemenea, vom memora $mx[i][j][k]$ care va ține dimensiunea maximă a unei submulțimi cu aceleași proprietăți. În funcție de cât de optimizată e soluția, poate trece și de subtaskul 3.

Pentru subtaskul 3 putem folosi aceeași soluție ca la subtaskul 2, deosebirea fiind că ținem cont doar de numerele care au un anumit bit egal cu 1. Astfel, dinamica se va transforma în $dp[i][j]$, $dp[i][j][k]$ e egal cu 1 dacă din primele i numere, putem alege numere astfel încât XOR-ul să fie egal cu j .

Soluția de 100 de puncte se bazează pe faptul că putem selecta toate numerele care au un bit anume setat, iar dacă XOR-ul lor e egal cu 0, putem exclude oricare dintre numere iar XOR-ul va deveni nenul. Pentru a ține XOR-urile pentru fiecare bit, vom procesa update-urile în $O(\log valmax)$ și vom răspunde la query-uri în $O(\log valmax)$.

2 Problema Connect the Tree

AUTOR: BOGDAN IOAN POPA

2.1 Descrierea soluției

Dificultate: Grea

Se liniarizeaza arborele. Fie $L[x]$ si $R[x]$ range-ul de valori intre care se gasesc nodurile din subarborele lui x .

Sa vedem cum tratam operatiile 1 $x y$. Sa presupunem ca y este tatal lui x . Daca se sterge muchia $x y$ atunci tot subarborele lui x se deconecteaza de arbore deci vom aduna la tot subarborele o valoare random.

Daca se readauga vom scadea valoarea random adunata la operatia anterioara de stergere.

Pentru operatiile de tip 2 se verifica daca cele doua noduri au aceeasi valoare.

Adunarea unei valori la un subarbore se poate realiza cu o structura de date cum ar fi aib sau aint pe liniarizarea arborelui.

3 Problema Sieve

AUTOR: GIORGIO AUDRITO

3.1 Descrierea soluției

Această problemă este un exercițiu în optimizarea unei soluții de tip brute-force folosind cât mai multe optimizări de constantă și de timp.

În primul rând, se pot memora legăturile dintre numere, evitând iterarea prin perechi inutile.

De asemenea, se pot găsi optimizări pe baza modului în care se ordonează numerele în vector.

Pentru obținerea punctajului maxim, se poate realiza o soluție de tip branch-and-bound, care reduce răspunsul și în același timp și numărul de iterații al algoritmului de tip brute-force.

Dificultate: Grea

4 Problema Words

AUTOR: WILLIAM DI LUIGI

4.1 Descrierea soluției

Dificultate: Ușoară

Această problemă a fost un exercițiu de convertire a șirului dat în baza 26. Pentru subtaskurile mici, se puteau precalculea toate răspunsurile iar pentru soluția de 100 de puncte trebuia avut grijă și la manevrarea numerelor mari, ceea ce se putea realiza cu ușurință folosind variabile de tip long long.

5 Problema Flappy Bird

AUTOR: STEFAN DASCALESCU

5.1 Descrierea soluției

Dificultate: Ușoară

Soluția acestei probleme constă în a verifica pentru fiecare pereche de intervale adiacente dacă intersecția lor nu e vidă, ceea ce se poate verifica foarte ușor folosind if-uri.

6 Problema Sorting Proximity

AUTOR: STEFAN DASCALESCU

6.1 Descrierea soluției

Dificultate: Grea

Putem considera vectorul ca fiind o listă de puncte într-un plan, unde punctul $(i, p[i])$ este punctul reprezentând al i -lea număr din vector.

Se poate observa cu ușurință ca nu este optim să schimbăm pozițiile i și j dacă $p[i] \leq p[j]$, așa că ne vom ocupa doar de celălalt caz.

Într-un schimb, vom considera dreptunghiul cu colțurile $(i, p[i])$ și $(j, p[j])$, astfel încât numărul total de inversiuni salvate este $x + y - 1$, unde x e numărul de puncte din interiorul dreptunghiului și y e numărul de puncte din interiorul sau de pe laturile dreptunghiului.

Pentru o poziție i , putem găsi o submulțime de candidați strict crescătoare, așa că se poate demonstra că dacă r e mai mic ca s , valoarea optimă a lui j pentru r e mai mică decât valoarea optimă a lui j pentru s . Astfel, putem folosi optimizarea divide and conquer pentru a rezolva problema în $O(n \log^2 n)$.

7 Problema Wine Tasting

AUTORI: GIORGIO AUDRITO

7.1 Descrierea soluției

Dificultate: Medie

Pentru 50 de puncte, se poate observa că vor exista $n - k + 1$ subsecvențe cu suma k , astfel va trebui să avem grijă la poziția corespunzătoare ce trebuie afișată.

Pentru următoarele 20 de puncte, putem memora toate sumele posibile, le sortăm și afișăm suma de pe poziția corespunzătoare.

Soluția de 100 de puncte constă într-o căutare binară pe răspuns. La un pas al căutării binare, vom verifica pentru numărul din algoritm câte sume sunt mai mici sau egale cu acel număr. Răspunsul va fi cel mai mic număr astfel încât sunt cel puțin k sume mai mici sau egale cu el.