Descrierea Soluțiilor
Concursul Național “InfoPro”, Runda 2
Grupa C1

1 Problema Numere

Propunător: prof. Dan Octavian Dumitrascu C.N. ”Dinicu Golescu”, Campulung Arges

Construim un vector $a[1]...a[n]$, in care $a[i]$ va reprezenta cati divizori proprii are i. Constructia este asemanatoare cu constructia ciurului lui Eratostene. Dupa realizarea vectorului, pentru punctul 1 este imediata rezolvarea, iar pentru punctul 2 vom construi un vector b in care vom memora numerele din $[1, N]$ care au exact T divizori proprii, aceste numere fiind puse crescator, apoi vom determina cate intervale au exact M numere cu T divizori proprii prin formula.

2 Problema Consolidare

Propunător: prof. Marius Nicoli, C.N. “Frații Buzești”, Craiova

Vom rezolva problema independent pentru fiecare rând de la 1 la H. Pentru rândul curent construim un vector S în care $S[i] = 1$ dacă avem zid în poziția i pe acel rând, și 0 în rest ($S[i] = 1$ dacă și numai dacă $V[i] \geq$ rând).

Observăm că lungimea bucății folosită pe un anume rând trebuie dividă lungimile tuturor secvențelor maximale formate cu elemente cu valoarea 0 în S, pentru acel rând. Evident că vom minimiza numărul de bucăți dacă folosim drept lungime cel mai mare divizor comun al lungimilor secvențelor.

3 Problema Splatoon

Propunător: Bogdan Iordache, Universitatea din București

3.1 Subtask 1

Pentru acest subtask, aproape orice implementare care simulează procesul descris în enunț ar trebui să obțină punctajul specificat.

Un exemplu ar fi: pentru fiecare unitate de timp, determinăm câte celule libere se colorează (ținând cont că după N unități de timp toate celulele libere sunt cu siguranță colorate, acest calcul se poate face cu ușurință fixând o unitate de timp t, apoi un jucător, iterând apoi prin toate celulele libere ale matricei și numărând câte dintre ele se află la distanța t de jucătorul fixat). Salvăm un vector C cu semnificația C_i numărul total de celule colorate după momentul i. Putem răspunde acum la fiecare interogare, parcurgând vectorul C.

Complexitate: $O(N^3M + QN)$

3.2 Subtask 2

Putem determina vectorul C de mai sus, mai eficient, astfel: pentru fiecare celulă liberă a matricei, determinăm care este cel mai apropiat jucător, astfel determinăm și timpul după care va fi colorată. Acest lucru îl facem fixând celula și iterând prin pozițiile jucătorilor. ($O(N^2M)$)

Pentru a răspunde la o interogare putem apela la două variante:

- pentru fiecare interogare P, căutăm binar primul moment de timp t, pentru care $C_t \geq P$. ($O(Q \log N)$)
- determinăm un vector R, cu semnificația: R_j este timpul minim după care cel puțin j celule sunt colorate. Acesta se poate determina ușor pe baza lui C ($R_j = t$ pentru orice $j \in [C_{t-1} + 1, C_t]$). Răspunsul pentru interogarea P este R_P. Ținând cont că R are cel mult N^2 poziții, complexitatea de răspundere la interogări este $O(N^2 + Q)$.

3.3 Subtask 3

Observăm că celulele aflate la distanța cel mult t de un jucător formează o submatrice pătratică de latură $2t + 1$ centrată în poziția jucătorului.

Tragem următoarea concluzie: celula (i, j) este colorată după cel mult t unități de timp, dacă există un jucător în submatricea $((i-t, j-t), (i+t, j+t))$.

Fie matricea A ce conține valoarea 1 pe pozițiile pe care se află jucători și 0 pe celelalte. Ne interesează să răspundem rapid la interogări de tipul: ”Se află cel puțin un jucător în submatricea X?”. Se poate răspunde la astfel de interogări în $O(1)$ folosind sume parțiale:

$$S_{i,j} = \sum_{1 \leq l \leq i, 1 \leq c \leq j} A_{l,c} = A_{i,j} + S_{i-1,j} + S_{i,j-1} - S_{i-1,j-1}$$

Având matricea S calculată, putem spune câți de 1 se află în submatricea $((i1, j1), (i2, j2))$ folosind formula:

$$S_{i2,j2} - S_{i1-1,j2} - S_{i2,j1-1} + S_{i1-1,j1-1}$$

Se conturează următorul algoritm:

- pentru fiecare celulă liberă încearcă fiecare moment de timp t, începând de la 0
- folosind matricea S determină dacă celula fixată este colorată la momentul t
- dacă da, contorizează această celulă în vectorul C

Pentru a răspunde la interogări, aplicăm metoda de la subtask-ul 2, odată ce determinăm vectorul C.

Complexitate: $O(N^2\bar{d} + Q \log N)$ sau $O(N^2\bar{d} + Q)$ în funcție de metoda aleasă pentru a răspunde la interogări (am notat cu $\bar{d}$ distanța medie de la fiecare celulă până la cel mai apropiat jucător).

3.4 Subtask 4

Dacă în soluția de la subtask-ul 3, în loc să fixăm iterativ momentul de timp t pentru fiecare celulă, îl căutăm binar, obținem complexitatea $O(N^2 logN + Q)$ sau $O(N^2 logN + QlogN)$, în funcție de metoda aleasă pentru a răspunde la interogări. Obținem astfel punctajul maxim.

O soluție mai eficientă se poate obține astfel:

- la momentul $t = 0$ consideră toate pozițiile de start ale jucătorilor. Numărul celor libere ne dă C_0. Marchează **toate** celulele de start.
- celulele accesate la timpul $t+1$ sunt celulele încă nemarcate, vecine pe cele 8 direcții cu celulele marcate la momentul t. Le determinăm, le marcăm pe toate, numărăm câte sunt libere și actualizăm C_{t+1}.

Întrucât fiecare celulă, este marcată doar o dată și accesată de un număr constant de ori (maxim 8, din fiecare vecin), putem determina timpul de colorare al tuturor celulelor în $O(N^2)$. Dacă folosim metoda a doua de răspundere la interogări, obținem complexitatea optimă de $O(N^2 + Q)$.

Echipa care a pregătit setul de probleme pentru această rundă a fost formată din:

- prof. Daniela Lica, Centrul Județean de Excelență Prahova, Ploiești
- prof. Flavius Boian, Colegiul Național "Spiru Haret" Târgu Jiu
- prof. Florentina Ungureanu, Inspectoratul Școlar Județean Neamț/ Colegiul Național de Informatică Piatra-Neamț

- prof. Marius Nicoli, Colegiul Național ”Frații Buzești” Craiova
- prof. Octavian Dumitrașcu, Colegiul Național ”Dinicu Golescu” Câmpulung Muscel
- stud. Bogdan Iordache - Universitatea din București
- stud. Florian Ușurelu - Universitatea din București
- stud. Stelian Chichirim - Universitatea din București