Olimpiada Națională de Informatică, Etapa națională, Clasa a VI-a
Descrierea soluțiilor

Comisia științifică

14 aprilie 2025

Problema 1. diff

Propusă de: prof. Dan Pracsiu - Liceul Teoretic "Emil Racoviță", Vaslui

Cerința 1 - 30p

Se utilizează un vector de frecvențe, notat cu fr, de lungime 10, în care $fr[i]$ reține numărul de apariții ale cifrei i în șirul a, $i = 1..9$. Parcurgem șirul a și fiecare cifră $a[i]$ o contorizăm în fr, iar la final aflăm valoarea maximă din fr.

Cerința 2 - 30p

Ca și la prima cerință, utilizăm vectorul fr. Parcurgem șirul a și la fiecare pas $i = 1..n$, îl punem pe $a[i]$ în vectorul de frecvențe. În acest moment în fr avem memorat numărul de apariții ale fiecărei cifre de la 1 la 9 din secvența $a[1], a[2], \ldots, a[i]$. Actualizăm diferența maximă dintre două valori nenule din fr.

Cerința 3 - 40p

Soluția 1

Avem două etape:

1. Să considerăm pentru început două cifre distincte $c1$ și $c2$. Dorim să determinăm diff-ul maxim (diferența maximă dintre numărul de apariții ale lui $c1$, minus numărul de apariții ale lui $c2$) care se obține dintr-o secvență din șir.

   Construim un vector d de lungime n.

   Parcurgem șirul a și la fiecare pas i avem cazurile:

   - $a[i] = c1$, atunci $d[i] = d[i-1] + 1$

- $a[i] = c2$, atunci $d[i] = d[i-1] - 1$
- $a[i] \neq c1$ și $a[i] \neq c2$, atunci $d[i] = d[i-1]$

Deci valorile din d cresc atunci când dăm peste cifra $c1$, scad când dăm peste $c2$ sau rămân nemodificate în cazul celorlalte litere. La fiecare pas i, vom reține în variabila mn cea mai mică valoare a lui d care s-a obținut până atunci când s-a găsit o cifră $c2$. Diferența maximă diff se actualizează cu valoarea $d[i] - mn$ de fiecare dată când întâlnim o cifră $c1$.

Atenție, valorile din d pot fi și negative sau zero, dar trebuie să ne asigurăm că valoarea mn s-a obținut atunci când am întâlnit cel puțin un $c2$.

2. Facem același algoritm pentru cele două cifre $c1$ și $c2$, parcurgând șirul de la dreapta la stânga.

Etapele 1 și 2 le efectuăm pentru orice cifre diferite, $1 \leq c1 \neq c2 \leq 9$. Numărul de pași la cerința 3 va fi deci $9 * 8 * n = 72 * n$.

Soluția 2

Rezolvarea cerinței este bazată pe algoritmul de subsecvență de sumă maximă (Kadane).

Pentru fiecare pereche de cifre $c1$ și $c2$, $1 \leq c1, c2 \leq 9$, $c1 \neq c2$ parcurgem șirul a și pentru fiecare element a_i al șirului vom defini variabila x astfel: 1 dacă $a_i = c1$, -1 dacă $a_i = c2$ sau 0 în caz contrar.

Vom adăuga valoarea x la suma subsecvenței curente, sumă ce reprezintă de fapt **diff**-ul maxim al subsecvenței pentru care cifra $c1$ are numărul de apariții maxim, iar cifra $c2$ are numărul de apariții minim.

Evident, vom reține valoarea maximă a sumei secvenței doar dacă cifra $c2$ apare.

```
for(int c1=1; c1<=9; c1++)
    for(int c2=1; c2<=9; c2++)
    {
        if (c1 == c2) continue;

        int ok = 0, x, s = 0;
        for(int i=1; i<=n; i++)
        {   ///alg. lui Kadane
            if (c[i] == c1) x = 1;
            else if (c[i] == c2) x = -1, ok = 1;
            else x = 0;

            s = s + x;
            if (ok) diff = max(diff, s);
            if (s < 0) s = 0, ok = 0;
        }
    }
```

Problema 2. prime

Propusă de: prof. Piț-Rada Ionel-Vasile - Colegiul Național "Traian", Drobeta-Turnu Severin

Soluția 1

Cerința 1 - 50p

Varianta 1-1 - $O(Q * N * sqrt(N))$

Parcurgem cele Q interogări și pentru fiecare pereche a, b vom număra numerele prime din secvența $a, a + 1, ..., b$. Punctajul obținut va depinde de modul cum verificăm primalitatea unui număr.

Varianta 1-2 - $O(N * sqrt(N) + Q * N)$

Verificăm primalitatea fiecărui număr din secvența $0...N$ și păstrăm informațiile într-un vector $v[i] = 1$, dacă i este prim, respectiv $v[i] = 0$ dacă i nu este prim. Parcurgem cele Q interogări și pentru fiecare pereche a, b vom număra numerele prime din secvența $a, a + 1, ..., b$.

Varianta 1-3 - $O(N * log(N) + Q * N)$

Utilizăm ciurul lui Eratostene și construim vectorul $ciur[i] = 0$, dacă i este prim, respectiv $ciur[i] = 1$, în caz contrar. Parcurgem cele Q interogări și pentru fiecare pereche a, b vom număra numerele prime din secvența $a, a + 1, ..., b$ însumând valorile.

$(1 - ciur[a]) + (1 - ciur[a + 1]) + ... + (1 - ciur[b])$

Varianta 1-4 - $O(N * log(N) + Q)$

Utilizăm ciurul lui Eratostene calculăm vectorul de sume parțiale $s[0...N]$. Parcurgem cele Q interogări și pentru fiecare pereche a, b vom număra numerele prime din secvența $a, a + 1, ..., b$ astfel: dacă $a = 0$, atunci numărul de numere prime este $s[b]$, în caz contrar ($a > 0$) numărul de numere prime este $s[b] - s[a - 1]$.

Cerința 2 - 50p

Varianta 2-1 - $O(Q * N * N * sqrt(N))$

Parcurgem cele Q interogări și în cadrul fiecărei interogări parcurgem toate secvențele $[a..b]$ cu $0 <= a <= b <= N$ și numărăm numerele prime în cadrul fiecărei secvențe.

Punctajul obținut va depinde de modul cum verificăm primalitatea unui număr și de modul cum efectuăm numărarea.

Varianta 2-2 - $O(N * log(N) + Q * N * N)$

Îmbunătățim varianta $2 - 1$ utilizăm ciurul lui Eratostene.

Varianta 2-3 - $O(N * log(N) + N * N + Q)$

Îmbunătățim varianta $2 - 2$ astfel încât să facem doar o singură dată parcurgerea secvențelor. Pentru aceasta folosim un vector de frecvență ce calculează numărul de secvențe cu p numere prime.

Parcurgem cele Q interogări și pentru fiecare interogări afișăm corespunzător frecvența cerută.

Varianta 2-4

Utilizând ciurul lui Eratostene calculăm vectorul $prime[]$ cu cele nr numere prime cuprinse în $[0..N]$, adăugăm vectorului elementele $prime[0] = -1$ și $prime[nr + 1] = N + 1$.

Se observă că pentru interogările în care se cere numărul secvențelor care nu conțin numere prime soluția se obține dacă numarăm pentru fiecare interval de valori $prime[i] + 1, ..., prime[i + 1] - 1$ câte secvențe $[a..b]$ avem cu $prime[i] + 1 <= a <= b <= prime[i + 1] - 1$, $0 <= i <= nr$.

Notăm cu $x = prime[i + 1] - prime[i] - 1$

Formula este $1 + 2 + ... + x = x * (x + 1)/2$ (obținută cu formula lui Gauss)

Pentru interogările în care se cere numărul secvențelor care conțin p numere prime, cu $p >= 1$, soluția este să calculăm numărul de secvențe $[a..b]$ cu $prime[i - 1] + 1 <= a <= prime[i]$ si $prime[i + p - 1] <= b <= prime[i + p] - 1$, cu $0 <= i <= nr + 1 - p$.

Vom nota cu $x = prime[i] - prime[i - 1]$ și cu $y = prime[i + p] - prime[i + p - 1]$

Formula acum se reduce la $x * y$

Trebuie însumate aceste produse pentru a obține rezultatul, care se memorează pentru a nu mai fi recalculat

$O(N * log(N) + pi(N) * min(Q, pi(N)))$, unde $pi(N)$ reprezintă numărul numerelor prime $<= N$.

Soluția 2

Construim cu ciurul lui Eratostene un vector caracteristic a, de lungime 50000, în care $a[i] = 1$, dacă i este număr prim sau $a[i] = 0$, dacă i nu este număr prim. Pe baza acestui vector construim apoi sumele parțiale sp, deci $sp[i] = a[0] + a[1] + \cdots + a[i]$.

Cerința 1 Complexitate O(N*log(log(N))+Q)

Pentru fiecare întrebare dată prin perechea (i, j), numărul de numere prime din intervalul $[i, j]$ este dat de $sp[j] - sp[i - 1]$. Atenție, dacă $i = 0$, atunci răspunsul este $sp[j] - sp[0]$.

Cerința 2 Complexitate $O(N * log(log(N)) + N * pi(N)) + Q)$, unde $pi(N)$ reprezintă numărul numerelor prime $<= N$

Pentru a înțelege mai bine algoritmul la această cerință, să vedem cum arată vectorii a și sp pentru numerele de la 0 la 16:

i = 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

———————————————————————

a = 0 0 1 1 0 1 0 1 0 0 0 1 0 1 0 0 0

sp = 0 0 1 2 2 3 3 4 4 4 4 5 5 6 6 6 6

Rețineți că cei doi vectori au de fapt lungimea 50000. Construim (precalculăm) încă doi vectori, fr și cnt, în care $fr[i]$ = câte valori din sp sunt egale cu i $cnt[i]$ = câte secvențe au exact i numere prime

Vectorul fr se construiește ușor, parcurgând sp. Cum construim însă pe cnt? Presupunem că suntem la un pas $i = 1..n$ și considerăm $x = sp[i]$.

Câte secvențe care se termină cu poziția i conțin zero numere prime? Răspunsul este dat de numărul de valori egale cu x obținute anterior. Să ne uităm la vectorul a obținut mai sus. La poziția 10 avem $x = sp[10] = 4$. Anterior mai avem încă trei de 4, deci sunt trei intervale care se termină cu 10 și au zero numere prime: $[8, 10]$, $[9, 10]$ și $[10, 10]$. Deci în $cnt[0]$ vom adăuga numărul de valori de x reținute anterior în vectorul fr.

Asemănător, pentru $x = sp[i]$, câte secvențe care se termină cu poziția i și au de exemplu două numere prime? Răspunsul este $fr[x - 2]$, care se adaugă la $cnt[2]$.

Ideea este deci că la fiecare pas, pentru $x = sp[i]$, pentru orice $j = 0..x$ putem contoriza câte intervale care se termină cu i au exact j numere prime, contorizând în $cnt[j]$ valoarea $fr[x - j]$. Nu uităm ca la final să-l adăugăm în fr pe x.

Problema 3. special

Propusă de: prof. Arișanu Ana-Maria - Colegiul Național "Mircea cel Bătrân", Râmnicu-Vâlcea

Considerații matematice

Numărul natural $\overline{ab}$ este special $\iff \overline{ab}^2 = \overline{xyz} \iff y = x + z \implies \overline{ab}^2 = 100 \cdot x + 10 \cdot (x + z) + z \implies \overline{ab}^2 = 110 \cdot x + 11 \cdot z \implies \overline{ab}^2 = 11 \cdot \overline{xz} \implies \overline{ab} \in \{11, 22, 33\}$

$\overline{ab} = 11 \implies \overline{ab}^2 = 121$

$\overline{ab} = 22 \implies \overline{ab}^2 = 485$

$\overline{ab} = 33 \implies \overline{ab}^2 = 1089$ care nu convine pentru că nu are 3 cifre.

Deci, doar 11 și 22 pot fi considerate numere speciale.

Un număr devine număr special prin eliminarea cel puţin a unei cifre dacă numărul are cel puțin 3 cifre și conține cel puțin două cifre de 1 sau cel puțin două cifre de 2.

Observație

Din cauza restricțiilor impuse soluția optimă nu poate fi obținută prin construirea efectivă a matricii.

Cerința 1 - 50p

Soluție de $35p$ - $O(N \times M)$

Se generează toate elementele matricii și se verifică, pentru fiecare număr format din două cifre, dacă acesta îndeplinește condiția de „număr special".

Soluție de $43p$ - $O(N \times M)$

Se calculează frecvența de apariție a numerelor 11 și 22, conform formulei date.

Soluție de $50p$ - complexitate $O(N + M)$

propusă de prof. Pit Rada Ionel-Vasile - Colegiul Național "Traian", Drobeta-Turnu Severin

Se observă că, pe fiecare linie a matricii variația rezultatului formulei de calcul este dată de termenul $4 * j$ (deoarece $15 * i + 2025$ rămâne constant pe linie).

Aplicând proprietatea modulo $(a+b)\%k = (a\%k+b\%k)\%k$ se precalculează resturile termenilor $(4 * j)\%k$, pentru j variind de la 0 la $M - 1$.

Pentru fiecare linie i, valoarea $x = (15 * i + 2025)\%k$ este constantă și ca urmare numerele 11, respectiv 22 nu pot apărea decât în coloanele j pentru care $(4 * j)\%k = 11 - x$, $(4 * j)\%k = 22 - x$ sau $(4 * j)\%k = (k - x + 11)$, $(4 * j)\%k = (k - x + 22)$.

Cerința 2 - 50p

Soluție de $30p$ - $O(N \times M)$

Se simulează deplasările construind matricile timpilor de acces la celule și se determină pe baza acestora punctele de întâlnire. Se contorizează doar punctele de întâlnire care au numere de cel puțin 3 cifre și care conțin fie cel puțin două cifre de 1, fie cel puțin două cifre de 2 (sau ambele).

Soluție de $50p$ - $O(max(gcd(N, M), log(min(N, M))))$

propusă de prof. Georgescu Alice - Colegiul Național "Mihai Viteazul", Ploiești

Un punct de întâlnire (lin, col) are următoarea proprietate într-o matrice de dimensiune $N \times M$ (indexată de la 0) $lin \cdot M + col = col \cdot N + lin \implies lin \cdot (M - 1) = col \cdot (N - 1) \implies lin/col = (N - 1)/(M - 1)$, cu $0 \le lin \le N - 1$, și $0 \le col \le M - 1$ (relația 1)

Aceasta înseamnă că toate punctele de întâlnire mențin un raport constant între indicele liniei și indicele coloanei, egal cu $(N - 1)/(M - 1)$.

Prin urmare, numărul punctelor de întâlnire este legat de $GCD(N - 1, M - 1)$ și coodonatele acestor puncte se pot calcula direct din relația 1.

Echipa

Problemele pentru această etapă au fost pregătite de:

- prof. Arișanu Ana-Maria - Colegiul Național "Mircea cel Bătrân", Râmnicu-Vâlcea
- prof. Georgescu Alice - Colegiul Național "Mihai Viteazul", Ploiești
- prof. Pintescu Alina - Colegiul Național "Gheorghe Șincai", Baia Mare
- prof. Opriță Petru-Simion - Liceul "Regina Maria", Dorohoi
- prof. Piț-Rada Ionel-Vasile - Colegiul Național "Traian", Drobeta-Turnu Severin
- prof. Pracsiu Dan - Liceul Teoretic "Emil Racoviță", Vaslui
- prof. Șerban Marinel-Paul - Colegiul Național "Emil Racoviță", Iași
- prof. Nodea Gheorghe-Eugen - Centrul Județean de Excelență Gorj
- stud. Gheorghieș Petruț-Rareș - Facultatea de Automatică și Calculatoare București
- stud. Răileanu Alin-Gabriel - Universitatea "Alexandru Ioan Cuza", Facultatea de Informatică, Iași