

Olimpiada Societății pentru Excelență și
Performanță în Informatică - etapa națională
Descrierea Soluțiilor
Clasa a VII-a

1 Problema Cat2Pal

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

Soluție brute-force – 35 puncte

Prima cerință se poate rezolva printr-o metoda brute-force. Putem itera X prin toate numerele de la 1 la $10 \cdot A$ și verifica prin construcția numerelor $X|A$ și $A|X$ (prin $|$ am notat operația de concatenare). Se pot obține astfel 20 puncte, în funcție și de constanta implementării.

Complexitate temporală: $\mathcal{O}(A \cdot \log(A))$

A doua cerință se poate rezolva tot folosind o metodă brute-force. Se pot analiza toate perechile de valori date. Se pot obține astfel 15 puncte, în funcție și de constanta implementării.

Complexitate temporală: $\mathcal{O}(N^2 \cdot \log(MaxVal))$

Soluție oficială – 100 de puncte

Pentru prima cerință se poate observa că este suficient a fi analizate:

- pentru concatenările $X|A$ prefixele oglinditului numărului A și prefixele concatenate la dreapta cu o cifră $\{0, 1, 2, \dots, 9\}$
- pentru concatenările $A|X$ sufixele oglinditului numărului A și sufixele concatenate la stânga cu o cifră $\{1, 2, \dots, 9\}$

Astfel se pot depune aceste numere într-un vector și apoi se verifică și se numără valorile distincte care corespund cerințelor.

Complexitate temporală: $\mathcal{O}(\log^2(A))$

Pentru a doua cerință se vor parcurge pentru fiecare valoare (asemănător cu rezolvarea de la prima cerință) prefixele și sufixele valorii oglindite și pentru fiecare dintre acestea se va încerca obținerea unui palindrom.

Pentru fiecare palindrom obținut se va construi un sufix identificator, pentru palindrom cu k cifre sufixul va avea $k - \frac{k}{2}$ cifre, care va fi marcat într-un vector de poziție. La final vor fi numărate pozițiile marcate.

Complexitate temporală: $\mathcal{O}(N \cdot \log^2(MaxVal))$

2 Problema Virus

Propunător: prof. Adrian Pintea, Colegiul Național "Andrei Mureșanu" Dej

Cerința 1 – 31 de puncte

Se vor determina numărul de mutații ale virusului pandemic existente în listă, mutații nu neapărat distincte după următorul algoritm:

- Citim codul virusul pe care îl notăm cu v și cu nr – numărul de mutații care inițial este 0.
- Citim codurile pe rând din fișier și comparăm fiecare cod citit cu v , numărând pozițiile pe care se afla caractere distincte. În cazul în care codul citit are aceeași lungime egală cu lungimea codului virusului v și diferă printr-o singură poziție, incrementăm nr .
- La final scriem numărul nr în fișierul de ieșire.

Complexitate temporală: $\mathcal{O}(N \cdot L_{max})$

Cerința 2 – 69 de puncte

Soluția brute-force – 16 puncte

Putem itera prin toate codurile date, iar dacă un cod este mutație, iterăm din nou prin toate codurile pentru a număra de câte ori apare această mutație în listă. Dacă găsim o mutație care apare de mai multe ori sau apare de la fel de multe ori ca cel mai bun răspuns de până acum, dar este mai mică din punct de vedere lexicografic, o actualizăm.

Complexitate temporală: $\mathcal{O}(N^2 \cdot L_{max})$

Soluția oficială

Pentru a obține o complexitate mai bună vom construi un tabel bidimensional fr cu maxim L linii (numărul de litere din codul virusului) și 52 coloane (asociate literelor mici și mari).

La citirea unui cod din fișier se verifică dacă este mutație și în caz afirmativ incrementăm $fr[pos][ch]$, unde pos este poziția pe care apare litera diferită față de virusul v , iar ch este litera prin care diferă față de virusul v .

Pentru determinarea maximului se parcurge tabelul bidimensional și se ține cont la modificarea valorii maxime și de ordinea lexicografică pentru mutație.

Complexitate temporală: $\mathcal{O}(N \cdot L_{max} + L_{max}^2 \cdot \Sigma)$

3 Problema Zid

Propunător: prof. Nistor Moț, Școala "Dr. Luca" Brăila

Vom memora configurația zidului într-o matrice Z cu N linii și M coloane, unde $Z[i][j]$ reprezintă culoarea cărămizii din poziția (i, j) în zid.

Soluția 1 – 22 de puncte

Parcurgem toate pătratele posibile și verificăm pentru fiecare pătrat dacă este uniform colorat, parcurgând pătratul și contorizând culorile care apar în pătrat într-un vector de frecvență.

Pentru a construi toate pătratele posibile fixăm colțul stânga-sus al pătratului în toate modurile posibile, apoi fixăm latura pătratului în toate modurile posibile (preferabil în ordine descrescătoare, astfel încât prima soluție găsită să fie de latură maximă).

Iterarea prin toate pătratele are complexitatea $\mathcal{O}(N \cdot M \cdot \min(N, M))$, iar pentru fiecare pătrat trebuie să facem și o verificare prin parcurgerea pătratului.

În final, **complexitatea temporală** este: $\mathcal{O}(N \cdot M \cdot \min(N, M) \cdot (N \cdot M + C))$.

Soluția 2 – 45 de puncte

Optimizăm soluția 1, încercând să evităm să parcurgem întreg pătratul pentru a calcula frecvențele culorilor.

Construim un tablou tridimensional Fr , unde $Fr[i][j][k]$ reprezintă numărul de apariții ale culorii k pe rândul i al zidului, considerând primele j cărămizi.

Precalculând acest tablou, putem afla frecvența de apariție a fiecărei culori într-un pătrat de latură L și colțul stânga-sus în poziția (i, j) parcurgând doar liniile de la i la $i + L - 1$ și determinând frecvențele de apariție ale culorilor pentru coloanele $j, \dots, j + L - 1$ prin scădere (pentru a determina frecvența de apariție a culorii k pe linia x scădem $Fr[x][j + L - 1][k] - Fr[x][j - 1][k]$).

Complexitatea temporală: $\mathcal{O}(N \cdot M \cdot \min(N, M) \cdot M \cdot C)$.

3.1 Soluția oficială – 100 de puncte

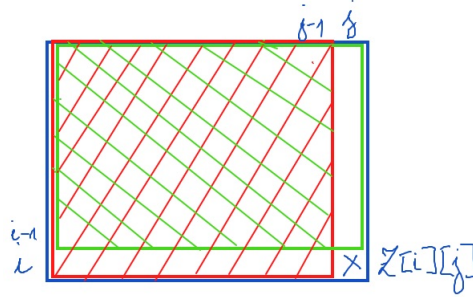
Optimizăm în continuare soluția 2, pentru a determina frecvența de apariție a unei culori într-un pătrat în complexitate $\mathcal{O}(1)$.

Pentru aceasta vom construi un tablou tridimensional Fr , unde $Fr[i][j][k]$ reprezintă numărul de apariții ale culorii k în submatricea cu colțul stânga-sus în poziția $(1, 1)$ și colțul dreapta-jos în poziția (i, j) . Precalcularea acestui tablou se poate face în $\mathcal{O}(N \cdot M \cdot C)$ astfel:

```

for  $i \leftarrow 1 \dots N$  do
  for  $j \leftarrow 1 \dots M$  do
    for  $k \leftarrow 0 \dots C$  do
       $Fr[i][j][k] = Fr[i-1][j][k] + Fr[i][j-1][k] - Fr[i-1][j-1][k] + (Z[i][j] == k)$ 

```

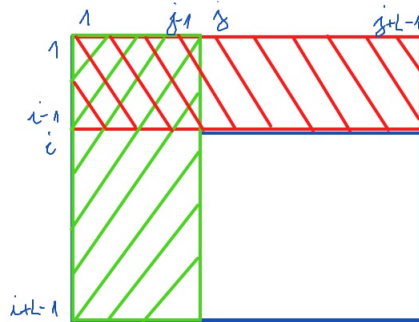


Pentru a memora frecvența de apariție a fiecărei litere într-un pătrat cu colțul în stânga-sus în poziția (i, j) și colțul din dreapta jos în poziția $(i+L-1, j+L-1)$ vom utiliza un vector uz , unde $uz[k]$ reprezintă frecvența de apariție a culorii k . Calcularea vectorului uz se poate face în $\mathcal{O}(C)$, în următorul mod:

```

for  $k \leftarrow 0 \dots C$  do
   $uz[k] = F[i+L-1][j+L-1][k] - F[i+L-1][j-1][k] - F[i-1][j+L-1][k] + F[i-1][j-1][k]$ 

```



Complexitatea temporală: $\mathcal{O}(N \cdot M \cdot \min(N, M) \cdot C)$.

Echipa

Setul de probleme pentru această rundă a fost pregătit de:

- prof. Emanuela Cerchez, Colegiul Național "Emil Racoviță" Iași
- prof. Nicoli Marius, Colegiul Național "Frații Buzești", Craiova
- prof. Adrian Pinte, Colegiul Național "Andrei Mureșanu" Dej
- prof. Ionel-Vasile Piț-Rada, Colegiul Național "Traian", Drobeta-Turnu Severin
- prof. Nistor Moț, Școala "Dr. Luca" Brăila
- student Theodor-Gabriel Tulbă-Lecu, Universitatea Politehnica București, Facultatea de automatică și calculatoare.