Olimpiada Societății pentru Excelență și Performanță în Informatică, Etapa Județeană

Descrierea Soluțiilor

Clasa a VIII-a

1 Problema Cartofi

Propunător: prof. Flavius Boian
Colegiul Național "Spiru Haret", Târgu Jiu

1.1 Soluție cu punctaj parțial

Cerința 1 (20 de puncte)

- O soluție cu punctaj parțial se poate obține generând șirul cifrelor unităților primilor $\mathbf{N} \cdot \mathbf{M}$ termeni ai șirului Fibonacci, contorizându-se apariția fiecărei cifre 0. Complexitate: $\mathcal{O}(\mathbf{N} \cdot \mathbf{M})$.

Cerința 2 (40 de puncte)

- O soluție cu punctaj parțial se poate obține utilizănd matricea $\mathbf{N} \cdot \mathbf{M}$ a numerelor cartofilor produși de plante. Se observă că suprafața pătratică de sumă maximă trebuie căutată printre submatricele $\mathbf{N} \cdot \mathbf{N}$ ale matricei. Astfel, suma numerelor din prima submatrice (formată din coloanele $(\mathbf{1}, \mathbf{N})$) este egală cu suma valorilor din primele $\mathbf{N}$ coloane. Submatricea formată din coloanele $(\mathbf{2}, \mathbf{N}+\mathbf{1})$ va avea suma egală cu suma valorilor din coloanele $(\mathbf{2}, \mathbf{N}+\mathbf{1})$. Această sumă se poate obține și direct, din suma anterioară scăzând suma primei coloane și adăugând suma coloanei $(\mathbf{N}+\mathbf{1})$. Procedăm asemănător pentru celelalte submatrice, ultima fiind formată din ultimele $\mathbf{N}$ coloane. Cea mai mare sumă dintre acestea reprezintă răspunsul la cerința 2.
Pentru a evita calculul repetat al sumelor valorilor din fiecare coloană, putem calcula mai întâi aceste sume într-un vector de sume ($SC[K]$ = suma valorilor din coloana K, $K = 1, 2, ..., \mathbf{M}$). Se observă inutilitatea formării matricei. Calculul sumei fiecărei coloane în vector se poate face simulând construirea matricei, adunând ultima cifră a termenului Fibonacci la suma ce îi corespunde. Complexitate $\mathcal{O}(\mathbf{M})$.

Cerința 3 (40 de puncte)

- O soluție cu punctaj parțial se poate utilizând matricea, calculând pentru fiecare pereche (A, B) suma elementelor situate între aceste coloane.

- O soluție cu un punctaj parțial mai mare ca anteriorul se obține utilizând un alt mod de calculare a sumelor elementelor cuprinse între coloanele (A, B) și constă în calculul sumelor parțiale ale vectorului SC intr-un alt vector S cu $\mathbf{M}$ componente. Astfel, suma elementelor dintre coloanele (A, B) va fi egală cu $S[B] - S[A-1]$, iar $S[0] = 0$. Complexitate $\mathcal{O}(\mathbf{Q} \cdot \mathbf{M})$.

1.2 Soluție cu punctaj maxim

Propusă de stud. Ioan-Cristian Pop (UPB) și stud. Ioan-Bogdan Iordache (UB)

General

- Știm că șirul cifrelor unităților termenilor din șirul lui Fibonacii este periodic cu perioada **60**. Fie $\mathbf{R} = \mathbf{60}$. Pe baza acestei proprietăți, putem construi strict primele **60** de elemente din acest șir, iar pe baza acestora putem calcula răspunsul la fiecare cerință.
  De exemplu, al **500**-lea termen din șir va fi egal cu al **20**-lea termen.

Cerința 1 (20 de puncte)

- Plecând de la observația de mai sus, construind vectorul alcătuit din primele **60** de numere din șir. De aici aflăm că cifra **0** apare de **4** ori în acest set. Știind că există $(\mathbf{N} \cdot \mathbf{M})/\mathbf{R}$ seturi complete, rămâne să determinăm numărul de apariții ale cifrei **0** în secvența formată din ultimii $(\mathbf{N} \cdot \mathbf{M})\%\mathbf{R}$ termeni ai vectorului.
  De exemplu, pentru $\mathbf{N} = \mathbf{50}$ și $\mathbf{M} = \mathbf{100}$, obținem $\mathbf{N} \cdot \mathbf{M} = \mathbf{5000}$. Știm că vor fi $[\mathbf{5000/60}] = \mathbf{83}$ seturi complete, iar în ultimii **20** de termeni, **0** se află o singură data. Deci, răspunsul va fi $\mathbf{83 \cdot 4 + 1 = 333}$.
- Complexitate: $\mathcal{O}(\mathbf{R})$.

Cerința 2 (40 de puncte)

- Este suficient să calculăm submatricea din colțul stânga sus (o numim submatricea **A**, de dimensiuni $\mathbf{min}(\mathbf{N}, \mathbf{60}) \cdot \mathbf{min}(\mathbf{M}, \mathbf{60}))$ și să calculăm primele $\mathbf{min}(\mathbf{M} - \mathbf{N} + \mathbf{1}, \mathbf{60})$ matrici de dimensiuni $\mathbf{N} \cdot \mathbf{N}$, începând de la stânga. Matricea are colțul stânga-sus în coordonatele $(\mathbf{1}, \mathbf{1})$.

- Ca să construim matricea, calculăm pentru fiecare poziție (i, j) din matrice al câtelea termen din șir este cu ajutorul formulei:

  - ☻ $(\mathbf{i} - \mathbf{1}) \cdot \mathbf{M} + \mathbf{j}$, pentru **i** impar;
  - ☻ $(\mathbf{i} - \mathbf{1}) \cdot \mathbf{M} + (\mathbf{M} - \mathbf{j} + \mathbf{1})$, pentru **i** par.

  De exemplu, pentru $\mathbf{N} = \mathbf{50}$, $\mathbf{M} = \mathbf{100}$, termenul de pe poziția $\mathbf{i} = \mathbf{20}$, $\mathbf{j} = \mathbf{30}$ este al $\mathbf{i} \cdot (\mathbf{N} - \mathbf{1}) + (\mathbf{M} - \mathbf{j} + \mathbf{1}) = \mathbf{1021}$-lea termen din șir, care la rândul lui este egal cu primul termen din șir (șirul fiind periodic cu perioada **60**).

- Pentru a construi eficient sumele parțiale pe coloane, determinăm frecvența de apariții a fiecărui termen din matricea **A** pe linii.

  De exemplu, dacă **N** ar fi egal cu **460**, termenul de pe prima linie se va repeta de **7** ori $((\mathbf{460} - \mathbf{1})/\mathbf{7})$, în timp ce termenul de pe linia **60** se va repeta de **6** ori $((\mathbf{460} - \mathbf{60})/\mathbf{7})$.

- Pentru a rezolva optim, calculăm inițial suma primei matrice, cu colțul stânga-sus în $(\mathbf{1}, \mathbf{1})$, pe baza sumelor parțiale pe coloane. O variantă ar fi să determinăm frecvența apariției fiecărei coloane în această matrice.

De exemplu, pentru $\mathbf{N} = \mathbf{79}$, frecvența primei coloane este **2**, în timp ce frecvența celei de-a **30**-a coloane este **1**.

Apoi, pentru a calcula suma matricei următoare, scădem suma de pe prima coloană și adăugam suma de pe coloana urmatoare (*sliding window*).

Complexitate $\mathcal{O}(\mathbf{R} \cdot \mathbf{R})$.

Cerința 3 (40 de puncte)

- O soluție cu punctaj mediu se poate obține în maniera descrisă la cerința 2, plecând de la sumele parțiale de pe coloane. Pentru fiecare pereche de coloane $(\mathbf{X}, \mathbf{Y})$ determinăm frecvența de apariție a fiecărei coloane dintre cele **60** și calculăm suma.

  Complexitate: $\mathcal{O}(\mathbf{Q} \cdot \mathbf{R})$.

- O soluție cu punctaj maxim, eficientă, presupune construirea unor sume parțiale pe baza sumelor parțiale de pe coloane (ideea poate fi aplicată și la cerința 2).

- Fie $\mathbf{sp[j]}$ = suma elementelor din primele j coloane. Apoi, pentru fiecare pereche de coloane $(\mathbf{X}, \mathbf{Y})$, aflăm de câte ori se repetă setul de **60** de coloane, și adaugăm restul de coloane neutilizate.

  De exemplu, pentru $\mathbf{X} = \mathbf{45}$ și $\mathbf{Y} = \mathbf{130}$ observăm că este un set complet de coloane care se repetă. Rămâne să adăugăm restul de coloane neutilizate: $\mathbf{45}, \ldots, \mathbf{60}$ și $\mathbf{121}, \ldots, \mathbf{130}$. Cu ajutorul sumelor parțiale, putem calcula imediat aceste sume: $\mathbf{sp}[\mathbf{60}] - \mathbf{sp}[\mathbf{44}]$, respectiv $\mathbf{sp}[\mathbf{10}] - \mathbf{sp}[\mathbf{0}]$.

- Astfel, raspundem in $\mathcal{O}(\mathbf{1})$ la fiecare întrebare. Complexitate finală: $\mathcal{O}(\mathbf{R} \cdot \mathbf{R} + \mathbf{Q})$ ($\mathbf{R} \cdot \mathbf{R}$ provine de la generarea matricei de $\mathbf{60} \cdot \mathbf{60}$, apoi construirea sumelor parțiale).

2 Problema Tunel

Autor:
prof. Cristina Sichim

Problema Tunel este ultima problemă propusă de regretata noastră colegă prof. Cristina Sichim. Problema a fost selectată pentru etapa națională a Olimpiadei de Informatică Gimnaziu - 2020, clasa a VII-a (ONIGim 2020), etapă anulată din cauza pandemiei. Selectarea și pregătirea acestei probleme pentru OSEPI 2021 reprezintă un omagiu pe care îl aducem celei care timp de două decenii ne-a fost o foarte apreciată și respectată colegă în comisiile de organizare a competițiilor naționale și internaționale de informatică din țara noastră.

2.1 Soluție cu punctaje variate

Soluția propusă utilizează o matrice binară A de dimensiune $(2\mathbf{N}-1)\cdot\mathbf{M}$, în care valorile de 1 din liniile impare reprezintă elementele unitare, iar valorile 1 din liniile pare reprezintă pasajele.

Observații: a) umplerea inutilă cu 1 a liniilor din matrice corespunzătoare tunelurilor va mări timpul de executare, diminuând punctajul; b) simularea deplasării în matrice se poate face din element în element sau din pasaj în pasaj.

Cerința 1 (40 de puncte)

- O soluție cu punctaj maxim (complexitate $\mathcal{O}(\mathbf{M})$) se obține pornind din elementul $A[2\mathbf{X}-1,1]$ corespunzător intrării în tunelul $\mathbf{X}$, simulând deplasarea după regulile date.
- Pentru fiecare intrare, traseul lui Tommy spre ieșire este unic.
- Astfel, din $\mathbf{A}[\mathbf{i},\mathbf{j}]$, Tommy se deplasează în: a) $\mathbf{A}[\mathbf{i+2},\mathbf{j+1}]$ dacă $\mathbf{A}[\mathbf{i+1},\mathbf{j}]=\mathbf{1}$ (există pasaj în jos) sau b) $\mathbf{A}[\mathbf{i-2},\mathbf{j+1}]$ dacă $\mathbf{A}[\mathbf{i-1},\mathbf{j}]=\mathbf{1}$ (există pasaj în sus) sau c) $\mathbf{A}[\mathbf{i},\mathbf{j+1}]$. Doar una din cele 3 situații a), b) și c) poate să apară.
- Exceptând tunelul cu eticheta $\mathbf{N-1}$ (căruia îi corespunde linia $\mathbf{2N-3}$ din matrice), deplasarea se finalizează în momentul în care se ajunge în coloana $j=\mathbf{M}$, valoarea indicelui i reprezentând numărul tunelului prin care Tommy iese.
- În cazul în care Tommy ajunge în ultimul element al tunelului cu eticheta $\mathbf{N}$, indiferent de valoarea $A[2\mathbf{N}-2,\mathbf{M}]$ (adică indiferent de existența unui pasaj către tunelul $\mathbf{N}-1$) Tommy ia recompensa si iese prin tunelul $\mathbf{N}$ cu recompensă.
- În cazul în care Tommy ajunge în ultimul element din tunelul $N-1$, dacă $A[2\mathbf{N}-2,\mathbf{M}]=1$ (adică există pasaj către ultimul tunel) atunci Tommy trece prin pasaj, ia recompensa si iese prin tunelul $\mathbf{N}$. Altfel, va ieși prin tunelul $\mathbf{N}-1$.

Cerința 2 (60 de puncte)

- O soluție cu punctaj partial și complexitate $\mathcal{O}(\mathbf{N}\cdot\mathbf{M})$ se obține folosind algoritmul descris la cerința 1 pentru fiecare dintre cele $\mathbf{N}$ tuneluri. În plus, vom contoriza fiecare element unitar și fiecare pasaj prin care vom trece. Valoarea acestui contor va reprezenta lungimea traseului parcurs. Răspunsul la cerința 2 va fi valoarea minimă a lungimilor traseelor cu ieșire prin tunelul $\mathbf{N}$.

- O soluție cu punctaj maxim se obține simulând deplasarea de la ieșire către intrare pentru cele maxim două trasee posibile. Plecând din $\mathbf{A[2N-1, M]}$, se trece în $\mathbf{A[2N-1, M-1]}$ sau în $\mathbf{A[2N-3, M]}$ (doar dacă $\mathbf{A[2N-2, M] = 1}$ adică există pasaj între ultimele elemente ale tunelurilor $\mathbf{N}$ și $\mathbf{N-1}$).

  Pentru fiecare dintre aceste aceste trasee vom proceda astfel: din $\mathbf{A[i, j]}$ trecem în a) $\mathbf{A[i-2, j-1]}$ dacă $\mathbf{A[i-1, j] = 1}$ sau b) $\mathbf{A[i+2, j-1]}$ dacă $\mathbf{A[i+1, j] = 1}$ sau c) $\mathbf{A[i, j-1]}$. Doar una dintre situațiile a), b) sau c) poate să apară. Astfel, sunt parcurse cel mult două trasee din cele N posibile (complexitate $\mathcal{O}(\mathbf{M})$).

2.2 Soluție cu punctaj maxim

Propusă de drd. Diana Ghinea (ETH Zürich) și ș.l. Stelian Ciurea (ULB Sibiu)

Complexitatea soluției, atât pentru cerința 1 cât și pentru cerința 2 este $\mathcal{O}(\mathbf{M})$ cu memorie suplimentară $\mathcal{O}(\mathbf{N} \cdot \mathbf{M})$, sau $\mathcal{O}(\mathbf{M} \log \mathbf{N})$ cu memorie suplimentară $\mathcal{O}(\sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{P_i})$, unde P_i reprezintă numărul de pasaje din tunelul i.

Cerința 1 (40 de puncte)

Soluția se poate obține simulând traseul lui Tommy element cu element de la intrarea dată până la ieșire. Notăm cu (T, E) elementul unitar cu numărul E din tunelul cu eticheta T. Orice traseu conține $\mathcal{O}(\mathbf{M})$ elemente întrucât fiecare element unitar al oricărui tunel este conectat la cel mult un pasaj. Pentru fiecare element $(i, j) \neq (\mathbf{N}, \mathbf{M})$, verificăm dacă există un pasaj spre tunelul $i-1$ sau spre tunelul $i+1$ din elementul (i, j). În caz afirmativ, Tommy va merge prin pasajul corespunzător către elementul $(i-1, j)$, respectiv $(i+1, j)$. Altfel, Tommy va continua să meargă către dreapta: fie în elementul $(i, j+1)$, fie iese din pasaj.

Putem verifica dacă există un pasaj din elementul $(i, j) \neq (\mathbf{N}, \mathbf{M})$ în mai multe moduri:

- Reținem o matrice `bool` P cu $\mathbf{N}$ linii și $\mathbf{M}$ coloane, unde $P[i][j] =$ `true` dacă și numai dacă există un pasaj din celula (i, j) către celula $(i+1, j)$. Astfel, putem verifica dacă există un pasaj în complexitate $\mathcal{O}(\mathbf{1})$, însă cu memorie suplimentară $\mathcal{O}(\mathbf{N} \cdot \mathbf{M})$.
- Utilizăm structura `vector` din C++: pentru fiecare coloană j, `vector`-ul $P[j]$ reține pasajele de pe coloana j (cel mult $\mathcal{O}(\mathbf{N})$ pasaje). Putem verifica dacă există un pasaj folosind căutare binară, în complexitate $\mathcal{O}(\log \mathbf{N})$, cu memorie suplimentară $\mathcal{O}(\sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{P_i})$, unde P_i reprezintă numărul de pasaje din tunelul i.
- Utilizăm structura `set` din C++: pentru fiecare coloană j, `set`-ul $P[j]$ reține pasajele de pe coloana j (cel mult $\mathcal{O}(\mathbf{N})$ pasaje). Putem verifica dacă există un pasaj (pentru elementul (i, j), dacă i aparține `set`-ului $P[j]$) în complexitate $\mathcal{O}(\log \mathbf{N})$, cu memorie suplimentară $\mathcal{O}(\sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{P_i})$, unde P_i reprezintă numărul de pasaje din tunelul i.

Astfel, utilizând prima modalitate, obținem complexitatea $\mathcal{O}(\mathbf{M})$, iar utilizând una dintre celelalte două modalități, obținem complexitatea $\mathcal{O}(\mathbf{M} \log \mathbf{N})$.

Cerința 2 (60 de puncte)

O soluție naivă ar simula traseul lui Tommy pornind din intrarea fiecărui tunel. Fără a lua în calcul timpul pentru verificarea pasajelor, o astfel de soluție ar avea complexitate $\mathcal{O}(\mathbf{N} \cdot \mathbf{M})$.

Pentru a obține o soluție în complexitatea de timp dorită, avem nevoie de următoarea **observație**:

Există cel mult două tuneluri din care Tommy poate ajunge la recomensă.

Orice traseu care ajunge la recompensă (îl vom denumi *traseu cu succes*) fie trece prin elementul $(\mathbf{N}-1, \mathbf{M})$ (dacă există un pasaj către $(\mathbf{N}, \mathbf{M})$), fie trece prin elementul $(\mathbf{N}, \mathbf{M}-1)$.

Presupunem că există cel puțin trei trasee distincte cu succes. Atunci, două dintre aceste trasee trec prin același element $(\mathbf{N}-1, \mathbf{M})$ sau $(\mathbf{N}, \mathbf{M}-1)$. Vom presupune că cele două trasee trec prin elementul $(\mathbf{N}-1, \mathbf{M})$ (celălalt caz este analog).

Vom nota aceste trasee cu $A = (a_l = (x, 1), a_{l-1}, \ldots, a_2 = (\mathbf{N}-1, \mathbf{M}), a_1 = (\mathbf{N}, \mathbf{M}))$ și $B = (b_{l'} = (x', 1), b_{l'-1}, \ldots, b_2 = (\mathbf{N}-1, \mathbf{M}), b_1 = (\mathbf{N}, \mathbf{M}))$ (acestee trasee nu includ pasajele).

Din moment ce A și B sunt doua trasee distincte, dar ultimele două elementul ale lui A și B sunt comune, există un element $C = a_k = b_k = (i_C, j_C)$ astfel încât $a_i = b_i$ pentru $1 \leq i \leq k$ și $a_{k-1} \neq b_{k-1}$. Acest lucru este posibil doar dacă unul dintre cele două trasee a trecut aici printr-un pasaj. Întrucât un element poate fi conectat la maxim un pasaj, doar unul dintre traseele A și B poate trece aici printr-un pasaj: presupunem ca acest traseu este A și că $a_{k+1} = (i_C - 1, j_C)$ (analog pentru celelalte cazuri). Atunci, $b_{k+1} = (i_C, j_C - 1)$ și, din moment ce aici este un pasaj, $b_{k-1} = (i_C - 1, j_C) \neq (i_C, j_C + 1) = a_{k-1}$, ceea ce contrazice modul în care am ales elementul C și demonstrează observația noastră.

Astfel, nu este necesar să simulăm toate traseele posibile pentru a obține răspunsul la această cerință: trebuie să simulăm cel mult două trasee.

Adăugăm încă o observație:

- *Există un traseu cu succes în care Tommy trece prin elementul* $(\mathbf{N}, \mathbf{M}-1)$.
- *Dacă există un pasaj între* $(\mathbf{N}-1, \mathbf{M})$ *și* $(\mathbf{N}, \mathbf{M})$, *atunci există un traseu cu succes în care Tommy trece prin elementul* $(\mathbf{N}, \mathbf{M}-1)$.

În concluzie, o soluție pentru această cerință va simula *în sens invers* deplasarea lui Tommy în cel mult două trasee (de la recompensă către intrare) și va calcula lungimea acestora:

- unicul traseu cu succes ce trece prin celula $(\mathbf{N}, \mathbf{M}-1)$, dacă acest traseu există (în cazul în care există un pasaj între celulele $(\mathbf{N}, \mathbf{M})$ și $(\mathbf{N}, \mathbf{M}-1)$);
- unicul traseu cu succes ce trece prin celula $(\mathbf{N}, \mathbf{M}-1)$.

Lungimea unui traseu poate fi calculată ca $\mathbf{M}+2p$, unde p este numărul de pasaje prin care Tommy trece în acel traseu.

Echipa

Setul de probleme pentru această rundă a fost pregătit de:

☺ prof. Flavius Dumitru Boian - Colegiul Național “Spiru Haret” , Târgu-Jiu

☺ prof. Veronica Raluca Costineanu - Colegiul Național “Ștefan cel Mare”, Suceava

☺ prof. Mirela Mlisan - Colegiul Național “Mircea cel Bătrân”, Râmnicu Vâlcea

☺ prof. Mircea Dorin Rotar - Colegiul Național “Samuil Vulcan”, Beiuș

☺ stud. Bogdan Ioan Iordache - Universitatea din București

☺ stud. Ioan Cristian Pop - Universitatea Politehnică București

☺ drd. Diana Ghinea - ETH Zürich

☺ prof. Carmen Mincă - Colegiul Național de Informatică “Tudor Vianu”, București

☺ ș.l. Stelian Ciurea - Universitatea “Lucian Blaga” din Sibiu