Olimpiada Națională de Informatică, Etapa județeană, Clasa a VIII-a Descrierea soluțiilor

Comisia științifică

March 16, 2025

Problema 1. Joc

Propusă de: stud. Dumitru Ilie, Facultatea de Matematică-Informatică, Universitatea București

Cerința 1 – O(N^3)

Vom fixa cele două valori st ($1 \leq st \leq n$) și v ($0 \leq v < st$) reprezentând începutul secvenţei şi vizibilitatea. Pentru fiecare secvenţă determinată de st şi v vom aplica unul dintre algoritmii liniari de calculare a elementului majoritar (resursă: Infoarena - Problema majorității votului). Astfel putem verifica, pentru fiecare secvență dacă este riscantă.

Cerința 1 – O(N^2)

Vom folosi un vector de frecvență, în care vom contoriza numărul de apariţii pentru fiecare decor. Să considerăm că am determinat vectorul de frecvenţă pentru secvenţa $[i,j]$ care începe la poziţia i şi se termină la poziţia j ($1 \leq i \leq j < n$). Când vom trece la secvența $[i,j+1]$ vom adăuga un singur element, deci putem actualiza ușor vectorul de frecvență. Elementul majoritar se poate recalcula în momentul în care adăugăm un element în vectorul de frecvență (fie rămâne valoarea precedentă, fie devine noua valoare adăugată). După ce toate secvențele cu capăt stâng i au fost analizate, resetăm vectorul de frecvență pentru a-l refolosi pentru subsecvențele cu capătul stâng $i + 1$.

Cerința 2 – O(N^2)

Vom simula efectiv jocul. Pentru a afla dacă o subsecvență este sau nu riscantă vom folosi un algoritm liniar de aflare a elementului majoritar (similar soluției de la cerința 1). Acestă soluție are complexitatea O(N^2) în cazul cel mai defavorabil.

Cerința 2 – O(N)

Pentru a optimiza determinarea elementului majoritar din soluţia precedentă, vom folosi un vector de frecvență, similar soluţiei 2 de la cerinţa 1. Să presupunem că ajungem pe subsecvența $[s - v,s]$ și știm dacă aceasta conține sau nu element majoritar, care este acesta și numărul său de apariții. Avem două cazuri:

- Secvența este riscantă. În acest caz știm că există element majoritar (fie acesta $emax$). Jucătorul va micșora vizibilitatea, astfel excluzând elementul de pe poziţia $s - v$. Datorită faptului că elementul $emax$ avea $(v + 1)/2 + 1$ apariții în subsecvența $[s - v,s]$, acesta rămâne elementul cu număr maxim de apariții și în subsecvența $[s - v + 1,s]$, deci ar fi posibil ca acesta să rămână element majoritar sau să nu mai existe element majoritar.

- Secvența nu este riscantă. În acest caz știm că nu există element majoritar. Când adăugăm un element în subsecvență avem două cazuri posibile. Fie acesta are acum numărul necesar de apariții, caz în care actualizăm $emax$, fie acesta nu are suficiente apariții, caz în care nu avem element majoritar.

Complexitatea acestei soluții este O(N) timp și O(N) memorie. Există și alte soluții, atât pentru punctaj integral, cât şi pentru punctaje parţiale.

Problema 2. Reducere

Propusă de: prof. Emanuela Cerchez, Colegiul Național "Emil Racoviță" Iași

Cerința 1.

Pentru a determina valoarea de reducere, trebuie să determinăm cel mai mare divizor comun al valorilor din secvență. Pentru a determina cmmdc pentru două valori utilizăm algoritmul lui Euclid. Pentru a determina cmmdc pentru o secvență de n valori utilizăm asociativitatea operaţiei cmmdc, deci determinăm la fiecare pas cmmdc dintre cmmdc-ul curent și următoarea valoare din secvență: $cmmdc(a_1, a_2, \cdots a_n) = cmmdc(...(cmmdc(cmmdc(a_1, a_2), a_3) \cdots a_n) \cdots)$.

Cerinţa 2.

Descompunem în factori primi fiecare valoare din secvenţă şi determinăm pe parcurs descompunerea în factori primi a celui mai mic multiplu comun al acestor valori (toţi factorii primi care apar în descompunerile valorilor din secvenţă la puterea cea mai mare). Factorii primi comuni la puterea cea mai mică constituie cmmdc (deci îi păstrăm în valoarea de reducere). Numărul minim de operaţii care trebuie să fie aplicate pentru a obţine valoarea de reducere este egal cu suma exponenţilor factorilor primi din descompunerea în factori primi a cmmmc/cmmdc.

Pentru subtask-ul 2, restricţiile permit utilizarea unui vector nr de 10^6 elemente, unde nr_i=puterea factorului prim i în descompunerea factori primi a cmmmc/cmmdc. Pentru a obţine punctele pe acest subtask nu este necesar să optimizăm descompunerea în factori primi utilizând pregenerarea numerelor prime cu ciurul lui Eratostene, dar, pentru subtask-ul 3, este necesar să descompunem în factori primi căutând divizorii, doar printre numerele prime până la radicalul numărului.

Pentru subtask-ul 3 restricțiile nu permit declararea vectorului *nr*. Ca urmare vom reţine o descompunere în factori primi ca o listă de factori primi şi puterile acestora, listă în care factorii primi apar în ordine crescătoare.

Pentru fiecare număr din secvenţă:

- descompunem numărul în factori primi;
- printr-un algoritm similar cu algoritmul de interclasare, actualizăm descompunerea în factori primi a cmmmc (în cmmmc trebuie să apară toți factorii primi la puterea cea mai mare).

La final simplificăm cmmmc cu cmmdc, parcurgând descompunerile în factori primi ale acestora şi, pentru factorii primi comuni, scăzând din puterea factorului prim din cmmmc puterea factorului prim respectiv din cmmdc.

Suma puterilor factorilor primi ai cmmmc după simplificarea cu cmmdc va fi numărul minim de operații de reducere necesare.

Echipa

Problemele pentru această etapă au fost pregătite de:

- Prof. Emanuela Cerchez, Colegiul Național "Emil Racoviță" Iași
- Stud. Dumitru Ilie, Facultatea de Matematică-Informatică, Universitatea București
- Prof. Nistor Moț, Liceul Teoretic "Dr. Luca" Brăila
- Prof. Isabela Patricia Coman, Colegiul Național de Informatică "Tudor Vianu" București
- Prof. Alin Burța, Colegiul Național "B. P. Hașdeu" Buzău
- Stud. Andrei Boacă, Facultatea de Informatică, Universitatea "Alexandru Ioan Cuza", Iași
- Stud. Victor Botnaru, Facultatea de Automatică și Calculatoare, Universitatea Națională de Știință și Tehnologie POLITEHNICA București
- Stud. Răzvan Alexandru Rotaru, Facultatea de Informatică, Universitatea "Alexandru Ioan Cuza", Iași
- Stud. Giulian Buzatu, Facultatea de Matematică-Informatică, Universitatea București
- Prof. Florentina Ungureanu, Colegiul Național de Informatică Piatra Neamț