

OLIMPIADA NAȚIONALĂ DE INFORMATICĂ, ETAPA JUDEȚEANĂ
CLASELE 11-12
DESCRIEREA SOLUȚIILOR

COMISIA ȘTIINȚIFICĂ

PROBLEMA 1: AVENTURA

Propusă de: Ivan Andrei-Cristian - Universitatea Politehnica București

Vom nota $k_1 + k_2 + \dots + k_N$ cu $\sum K$.

Subtask 1: Se generează toate permutările șirului $1, 2, \dots, N$, iar pentru fiecare șir se încearcă completarea nivelelor în ordinea generată. Complexitate $O(N! \cdot \sum K)$.

Subtask 2: O dependență circulară de dimensiune exact 2 reprezintă, în graful format de restricțiile jocului, un ciclu de lungime 2. Mai precis, dacă fixăm două nivele i și j din graf și există muchii în ambele sensuri, atunci este clar că ambele nivele nu vor putea fi completate vreodată. Acest lucru se poate verifica ușor dacă muchiile din graf sunt reținute într-o matrice de adiacență. Putem porni din unul dintre cele două nivele o parcurgere DFS, deoarece orice nivel care anterior era condiționat de unul dintre aceste nivele clar nu va fi parcurs niciodată. În final, restul nivelelor nemarcate vor putea fi completate în joc. Complexitate $O(N^2)$.

Subtask 3: Pentru fiecare nivel, vom ține minte $grad[x]$ – numărul de nivele de care este condiționat nivelul x . La fiecare pas, vom căuta un nivel i care are $grad[i] = 0$ și vom scădea gradul fiecărui nivel condiționat de nivelul i . Complexitate $O(N^2 + \sum K)$.

Subtask 4: Vom folosi construcția de la subtaskul precedent, însă, în loc să căutăm manual nivelele cu gradul 0, vom utiliza o coadă. Mai întâi, vom adăuga în coadă toate nivelele care au gradul inițial egal cu 0. Apoi, vom parcurge coada, iar în momentul în care scădem gradele vecinilor nivelului curent, dacă unul dintre vecini ajunge la gradul 0, îl vom adăuga în coadă. Complexitate $O(N + \sum K)$.

PROBLEMA 2: CROMATIC

Propusă de: profesor Szabó Zoltan - Colegiul "Petru Maior" Reghin

Să presupunem că șirul $a = (a_1, a_2, \dots, a_n)$ este cromatic. Condiția că șirul de intervale închise $minmax = ([min_1, max_1], [min_2, max_2], \dots, [min_n, max_n])$ are toate elementele distincte implică că orice element a_k , dacă este comparat cu intervalul de pe poziția anterioară lui k : $[min_{k-1}, max_{k-1}]$, atunci $a_k < min_{k-1}$ sau $a_k > max_{k-1}$.

De aici putem deduce că, față de elementul a_1 , elementele următoare din șir formează două subșiruri strict monotone. Elementele mai mari decât a_1 formează un șir strict crescător, iar elementele mai mici decât a_1 formează un șir strict descrescător. Dacă aceste două subșiruri le interclasăm în toate modurile posibile, fiecare șir nou obținut va fi un șir cromatic.

Considerând că avem un șir $a = (a_1, a_2, \dots, a_n)$, nu neapărat cromatic, pentru a obține prin rearanjarea elementelor toate permutările cromatice, vom rearanja elementele șirului în ordine crescătoare.

În continuare vom considera că elementele șirului a sunt ordonate crescător.

Se poate demonstra ușor că **dacă în șir există două elemente egale**, atunci șirul nu este cromatic, și nu există nicio altă permutare cromatică, deci $NSC = 0$.

Pentru a calcula numărul șirurilor cromatice NSC , în continuare vom accepta că șirul a este

strict crescător: $a_1 < a_2 < \dots < a_n$.

Acest șir este cromatic, și în lista șirurilor cromatice este primul în ordine lexicografică.

Observăm că în lista ordonată lexicografic a tuturor șirurilor cromatice elementele respectă următoarele proprietăți:

(Grupa 1) $a_1, a_2, a_3, \dots, a_n$

(Grupa 2) a_2 urmat de (a_3, a_4, \dots, a_n) și (a_1) interclasate în toate modurile posibile

(Grupa 3) a_3 urmat de (a_4, a_5, \dots, a_n) și (a_2, a_1) interclasate în toate modurile posibile

⋮

(Grupa k) a_k urmat de $(a_{k+1}, a_{k+2}, \dots, a_n)$ și $(a_{k-1}, a_{k-2}, \dots, a_1)$ interclasate în toate modurile posibile

⋮

(Grupa n) a_n, a_{n-1}, \dots, a_1

Observăm că în fiecare grupă k ($1 \leq k \leq n$) numărul șirurilor cromatice distincte este egal cu C_{n-1}^{k-1} .

De aici deducem că $NSC = C_{n-1}^0 + C_{n-1}^1 + \dots + C_{n-1}^{n-1} = 2^{n-1}$

În cadrul unei grupe k , ordinea lexicografică a șirurilor este determinată de elementele subșirului descrescător $(a_{k-1}, a_{k-2}, \dots, a_1)$. Numărul de modalități de aranjare a celor $k-1$ de elemente pe cele $n-1$ poziții este egal cu C_{n-1}^{k-1} . Folosind această proprietate vom putea calcula poziția p a unui șir în lista permutărilor cromatice, sau pentru un șir cromatic dat poziția lui q în ordine lexicografică, folosind sumă de combinații de ordin tot mai mic.

Subtaskurile 1 și 2: Cazul $c = 1$, numărul șirurilor cromatice va fi egal cu $NSC = 0$, dacă avem elemente egale în șir, respectiv $NSC = 2^{n-1}$ modulo 1 000 000 007.

Subtaskurile 3, 4, 5, 6: Cazul $c = 2$,

În subtaskul 3 este vorba despre o poziție p dintre primele n soluții în ordine lexicografică. Această cerință se poate rezolva cu un ciclu simplu în complexitate $O(n)$.

În subtaskul 4 este vorba despre o poziție p dintre ultimele n soluții în ordine lexicografică. Și această cerință se poate rezolva cu un ciclu simplu în complexitate $O(n)$.

Următoarele subtaskuri tratează cazuri generale: În subtaskul 5 avem $n \leq 20$, și se poate rezolva cu un algoritm backtracking de complexitate $O(2^{n-1})$.

Subtaskul 6 se rezolvă în complexitate $O(n)$ cu ajutorul sumelor de combinații.

Subtaskurile 7, 8, 9, 10: Cazul $c = 3$,

Similar cu subtaskurile 3, 4, 5 și 6, observăm că: Subtaskul 7, fiind caz particular se rezolvă în complexitate $O(n)$. Subtaskul 8, fiind caz particular se rezolvă în complexitate $O(n)$. Subtaskul 9, se poate rezolva cu un algoritm backtracking de complexitate $O(2^{n-1})$. Subtaskul 10 se rezolvă în complexitate $O(n)$ cu ajutorul sumelor de combinații.

PROBLEMA 3: EXPERIMENTE

Propusă de: Andrei Grigorean - CS Academy

Observație inițială: Problema poate fi reformulată astfel: dându-se un șir circular și update-uri sub forma unor intervale continue, să se calculeze după fiecare update cardinalul intersecției tuturor intervalelor de până atunci.

Subtask 1: Restricțiile pentru acest subtask implică faptul că putem considera șirul ca fiind necircular. În acest caz intersecția unor intervale va fi tot timpul un singur interval continuu. Este de ajuns să reținem cel mai mare capăt de start și cel mai mic capăt de final al intervalelor de update pentru a afla răspunsul.

Subtaskurile 2, 3, 4, 5: Pentru celelalte subtaskuri, intersecția update-urilor va fi reprezentată de o mulțime M formată din mai multe intervale, nu doar unul singur (inițial considerăm M formată dintr-un singur interval $[1, N]$). În momentul în care avem un update nou U , pentru fiecare interval I din M există 3 scenarii posibile:

- (1) I este inclus complet în U , trebuie păstrat în M ;
- (2) I nu se intersectează cu U , trebuie scos din M ;
- (3) I și U se intersectează fără ca I să fie complet inclus în U , I trebuie înlocuit.

Pentru scenariul (3), facem următoarele observații:

- există maxim 2 intervale I care se pot afla în această situație;
- I trebuie înlocuit fie cu un alt interval, fie cu alte două intervale.

În funcție de structura de date aleasă, se pot rezolva diferite subtaskuri. Soluția oficială folosește un set pentru o complexitate de $O(M \log M)$.

ECHIPA

Problemele pentru această etapă au fost pregătite de:

- Szabó Zoltan, Inspectoratul Școlar Județean, Târgu-Mureș
- Grigorean Andrei, CS Academy, București
- Ivan Andrei-Cristian, Universitatea Politehnica, București
- Argherie Ovidiu-Alexandru, Delft University of Technology
- Bogdan Vlad-Mihai, Universitatea București
- Ciorte Liviu, Pexabit, București
- Constantinescu Andrei-Costin, ETH Zurich
- Feodorov Andrei, ETH Zurich
- Floare Doru, Universitatea "Babeș-Bolyai", Cluj Napoca
- Moca Andrei Cătălin, Universitatea "Babeș-Bolyai", Cluj-Napoca
- Oncescu Costin-Andrei, Harvard University
- Popescu Adrian Andrei, Universitatea Politehnica, București
- Posdărăscu Eugenie Daniel, Youni, București
- Sillion Liviu-Mihai, Universitatea "Babeș-Bolyai", Cluj-Napoca
- Stănescu Matei-Octavian, Universitatea Politehnica, București
- Todoran Alexandru-Raul, Harvard University
- Tinca Matei, VU Amsterdam