Descrierea Soluțiilor
Olimpiada Societății pentru Excelență și Performanță în Informatică,
Etapa Națională
Baraj Seniori 1

1 Problema GP

Propunător: Ionel-Vasile Piț-Rada; pregătitor principal: Alexa Tudose

Fie A șirul elementelor mutate pană acum printr-o operație care inserează la începutul raftului. Similar, fie B șirul elementelor mutate printr-o operație care inserează la sfârsit. Inițial, A și B sunt vide. Vrem ca la final să maximizăm lexicografic $A + B$, unde $+$ semnifică operația de concatenare a două șiruri de numere.

Spunem că $p[i]$ este *maxim pe prefix* dacă pentru orice $j < i$ avem $p[j] < p[i]$. Similar, spunem că $p[i]$ este *maxim pe sufix* dacă pentru orice $j > i$ avem $p[j] < p[i]$.

Observație 1. *Pentru a obține soluția optima, A trebuie să fie descrescător la final, ceea ce înseamnă că trebuie să fie descrescător la orice moment.*

Demonstrație. Să presupunem că în soluția optimă A nu este descrescător la final. Fie k o poziție astfel încât $A[k] < A[k+1]$. Dacă am fi pus elementul $A[k]$ în șirul B in loc să îl punem în A, am fi obtinut o soluție mai bună (deci contradicție!). □

Observație 2. *Să considerăm o soluție intermediară $A + B$ în care elementele mutate sunt $p[1 \dots x] \cup p[y \dots N]$. Pentru a obține soluția optimă la final, $A + B$ trebuie să fie maxim lexicografic între toate șirurile $A' + B'$ ce se pot obține dacă mutăm $p[1 \dots x] \cup p[y \dots N]$.*

Observație 3. *Pentru a obține soluția optimă la final, nu avem voie să punem un element $p[i]$ în A dacă nu este nici maxim pe prefix, nici maxim pe sufix.*

Demonstrație. Fie i, j, k astfel încât $j < i < k$, $p[j]$ e primul maxim pe prefix la stânga lui i, iar $p[k]$ e primul maxim pe sufix la dreapta lui i. Presupunem că soluția optimă se obține dacă îl inseram pe $p[i]$ în A. Atunci când i devine activ (devine capăt și poate fi pus în A sau B), j sau k (poate chiar amândouă) a fost mutat deja în A sau B.

Să presupunem că j a fost mutat deja (celălalt caz este asemănător). Datorită observației 1, j a fost sigur pus în B (pentru că $p[i] < p[j]$). Dar noi puteam să obținem o soluție intermediară mai bună, folosind aceleași elemente din p, dacă puneam doar $p[j]$ în șirul A și restul în șirul B (construcția noastră are $p[i]$ ca prim element în A, iar cealaltă are $p[j]$ ca prim element). Datorita observatiei 2, obtinem o contradictie! □

Primul pas Să fixăm prima mutare. Sunt două variante: punem $p[1]$ sau $p[N]$ ca prim element pe raftul nou. Notăm cu *start* valoarea inițială ($p[1]$ sau $p[N]$) și cu r șirul obținut din p prin eliminarea lui *start*.

Observație 4. *Nu este optim să punem în A un element $r[i]$ dacă $r[i] < start$.*

Observație 5. *Este mereu optim să punem în A toate elementele din r care sunt maxim pe prefix sau sufix (în r) și care sunt mai mari decat start.*

Elemente speciale: Fie $special(i) = r[i]$ e maxim pe prefix sau maxim pe sufix și $r[i] > start$.

Solutie: Începem cu i=1 și j=N-1. La fiecare pas vom muta fie elementul $r[i]$, fie elementul $r[j]$, după cum urmează:

- Dacă $special(i)$ și $special(j)$: Îl punem în A pe cel mai mic
- Dacă $special(i)$ și $!special(j)$:
  - Dacă $r[i]$ poate fi pus în A (toate elementele speciale mai mici decât el au fost puse deja), atunci îl punem pe $r[i]$ în A
  - Altfel îl punem pe $r[j]$ în B
- Dacă $!special(i)$ și $!special(j)$: Îl punem în B pe cel mai mare

2 Problema ArbSumPow

Propunător: Tamio-Vesa Nakajima

Soluție in $O(2^N)$. Pentru primul subtask se poate folosi o soluție tip backtracking – se enumera efectiv toți subarborii, se calculează valoarea lor, și se însumează.

Soluție când $P = 1$. Pentru subtaskurile de acest fel, se observă că soluția este egală cu suma pentru fiecare nod i din $v(i)$ înmulțit cu numărul de subarbori care îl conțin pe i. Această valoare poate fi calculată cu ajutorul programării dinamice pe arbore, fie in $O(N^2)$, fie în $O(N)$.

Soluție când $P = 2$. În mod asemănator cu subtaskurile anterioare, se observă că soluția este egală cu suma pentru fiecare pereche (i, j) de noduri din $v(i)v(j)$, înmulțită cu numărul de subarbori care conțin și pe i și pe j. Asemănator cu subtaskul anterior, această valoare poate fi calculată în $O(N)$ sau $O(N^2)$ cu programare dinamică pe arbore.

Soluție de 100 de puncte. Vom calcula dinamica $d[i][j]$, unde $i \in \{1, \ldots, N\}$ și $j \in \{0, \ldots, P\}$, $d[i][j]$ este suma, pentru orice subarbore S care are pe i ca rădăcină, din $\left(\sum_{x \in S}\right)^j$. Este ușor de văzut că soluția este $\sum_{i=1}^{N} d[i][P]$. Prima observație este că, pentru i o frunză, $d[i][j] = v[i]^j$. Observăm apoi că, dacă $d[i][j]$ reprezintă valorile din dinamică considerând doar primii t fii ai lui i, iar k este al $t+1$-lea fiu al lui i, trebuie să reînoim dinamica în felul următor:

$$d'[i][j] = d[i][j] + \sum_{x=0}^{j} \binom{j}{x} d[i][x] d[k][k-x].$$

În această expresie d' reprezintă valoarea ce ar trebui să o aibă d ca să țină cont și de k. Observăm că această reînoire se poate calcula în $O(P^2)$.

Astfel, pentru a calcula dinamica pentru toate nodurile, arborele se parcurge DFS, dinamica se inițializează de parcă ar fi o frunza, iar apoi se iau în considerare fiii săi pe rând cu reînnoirea descrisă anterior. Cum reînnoirea se face odata pentru fiecare muchie din arbore, rezultă că complexitatea finală este $O(NP^2)$.

3 Problema Sezon

Propunător: Andrei Constantinescu

Pentru a verifica dacă șirul v dat de Marcel este unul valid (adică dacă rspunsul pe scenariu este `'1'`), se verifică mai întâi dacă acesta nu conține valori repetate (caz în care răspunsul este clar `'0'`). Dacă se trece de această verificare, se identifică apoi rădăcina arborelui cu nodul de valoare v maximă (nodul unde se află Marcel în săptămână M). Dacă, înrădăcinând astfel arborele, există 2 noduri vecine în arbore a, b cu proprietatea că a este părintele lui b și $v_a < v_b$, atunci răspunsul pe scenariu va fi `'0'`. Dacă se trece și de aceasta verificare, se sortează nodurile arborelui crescător după valorile lor v și se parcurg în ordine. Pentru fiecare pereche de noduri adiacente în parcurgere, să zicem a, b (unde a apare primul în parcurgere), mai verificăm și următoarele condiții:

1. $v_b - v_a \geq dist(a, b)$, unde prin $dist(a, b)$ am notat distanța dintre nodurile a și b în arborele nostru.

2. $v_b - v_a$ și $dist(a, b)$ au aceeași paritate.

Aceste verificări necesită calculul rapid al distanțelor în arbore, lucru ce se poate realiza în complexitate $O(N \log N)$ folosind orice algoritm standard de LCA (*Lowest Common Ancestor*). Dacă aceste verificări nu sunt îndeplinite, atunci răspunsul pe scenariu va fi, din nou, `'0'`. Ultima condiție care trebuie verificată este dacă rădăcină are un vecin al cărei valoare v este cu exact 1 mai mică decât a rădăcinii (nodul unde Marcel se află în săptămână $M - 1$), caz în care răspunsul este `'1'`, și altfel `'0'`.

Echipa

Setul de probleme pentru această rundă a fost propus și pregătit de:

- Adrian Panaete, profesor Colegiul "A. T. Laurian", Botoșani
- Alexandra Udriștoiu, studentă Universitatea din București
- Alexandru Petrescu, student Universitatea Oxford, Keble College
- Alexa Tudose, studentă Universitatea Oxford, University College
- Andrei Arhire, Universitatea "Alexandru Ioan Cuza" din Iași
- Andrei-Costin Constantinescu, student Universitatea Oxford, Balliol College
- Bogdan Ciobanu, Hudson River Trading.
- Bogdan Ioan Popa, student Universitatea din București.
- Bogdan Iordache, student Universitatea din București
- Bogdan Pop, student Universitatea "Babeș-Bolyai"
- Bogdan Sitaru, student Universitatea Oxford, Hertford College
- Costin-Andrei Oncescu, student Universitatea Oxford, St. John's College
- Emanuel Dicu, student Universitatea Politehnica Bucuresti
- Gabriel Tulbă-Lecu, student Universitatea Politehnica București
- George Chichirim, student Universitatea Oxford, Keble College
- Ionel-Vasile Piț-Rada, profesor Colegiul Național "Traian" Drobeta Turnu Severin
- Lucian Bicsi, doctorand Universitatea din București.
- Marius Nicoli, profesor Colegiul Național "Frații Buzești" Craiova
- Mihai Popescu, student Universitatea "Babeș-Bolyai"
- Stelian Chichirim, student Universitatea din București
- Tamio Vesa Nakajima, student Universitatea Oxford, University College
- Vlad Gavrilă, Google.
- Zoltan Szabo, profesor Liceul Tehnologic "Petru Maior" Reghin / ISJ Mureș Tg. Mureș