

# Descrierea soluției pentru problema Meet

Problemă propusă de Bogdan Pop

August 2023

## Introducere

În această problemă se dă un arbore cu  $N$  noduri, și  $Q$  interogări. Pe parcursul interogărilor se menține o mulțime  $S$  de noduri, inițial goală. Fiecare interogare fie adaugă sau scoate un nod din mulțimea  $S$ . După fiecare interogare trebuie să găsim numărul de noduri echidistante față de toate nodurile din  $S$ .

## A Subtaskul 1

În acest subtask, se garantează că  $N, Q \leq 300$ . Vom da o soluție cu complexitatea  $O(QN^2)$ . Mai întâi, folosind  $N$  parcurgeri în adâncime, vom calcula distanța între oricare două noduri. Apoi, la fiecare interogare, încercăm unul câte unul fiecare nod, și vedem dacă este echidistant față de toate nodurile din  $S$ . Cum  $|S| \leq N$ , și încercăm maxim  $N$  noduri, vedem că o interogare este rezolvată în complexitatea  $O(N^2)$ , de unde rezultă complexitatea dorită.

## B Subtaskul 2

Fie  $S = \{v_1, \dots, v_k\}$ . Observăm că un nod  $x$  este echidistant față de toată mulțimea  $S$  dacă și numai dacă  $\text{dist}(x, v_i) = \text{dist}(x, v_{i+1})$  pentru  $1 \leq i < k$ . Cum arată mulțimea de noduri  $x$  pentru care  $\text{dist}(x, v_i) = \text{dist}(x, v_{i+1})$ ? Considerăm lanțul dintre  $v_i$  și  $v_{i+1}$ : dacă el are lungime impară atunci mulțimea e vidă, altfel fie  $u_i$  nodul la mijlocul acestui lanț. Un nod  $x$  satisface  $\text{dist}(x, v_i) = \text{dist}(x, v_{i+1})$  dacă și numai dacă nodul cel mai apropiat de  $x$  din lanțul de la  $v_i$  la  $v_{i+1}$  este  $u_i$ . Această mulțime poate fi văzută în felul următor: dacă arborele ar fi înrădăcinat la  $u_i$ , atunci ar fi eliminate din mulțime exact subarborii care îi conțin pe  $v_i$  și  $v_{i+1}$ .

Parcurgem acum arborele în adâncime, și ordonăm nodurile linear în funcție de ordinea parcurgerii. Fie  $\text{last}(x)$  ultimul nod din subarborii lui  $x$  pe care îl vizităm. Observăm că mulțimea nodurilor care satisfac  $\text{dist}(x, v_i) = \text{dist}(x, v_{i+1})$  este formată dintr-un număr constant de intervale noduri, dacă acestea sunt considerate în ordinea dată de linearizare. Mai exact, dacă  $s$  este primul nod în linearizare, și  $t$  este ultimul nod în linearizare, avem următoarele cazuri.

$\text{lca}(v_i, v_{i+1}) = u_i$ . Presupunem fără pierdere de generalitate că  $v_i$  e parcurs înainte de  $v_{i+1}$ . În acest caz, fie  $a$  fiul lui  $u_i$  în a cărui subarbore se află  $v_i$ , și  $b$  fiul lui  $u_i$  în a cărui subarbore se află  $v_{i+1}$ . Submulțimea de noduri pentru care  $\text{dist}(x, v_i) = \text{dist}(x, v_{i+1})$  este  $[s, a) \cup (\text{last}(a), b) \cup (\text{last}(b), t]$ .

$\text{lca}(v_i, v_{i+1}) \neq u_i$ . În acest caz observăm că  $u_i$  trebuie să fie strămoșul unuia dintre  $v_i$  sau  $v_{i+1}$ . Presupunem fără pierdere de generalitate că  $u_i$  este strămoșul lui  $v_i$ . Fie  $a$  fiul lui  $u_i$  în a cărui subarbore se află nodul  $v_i$ . Submulțimea dorită este acum  $[u_i, a) \cup (\text{last}(a), \text{last}(u_i)]$ .

LCA-urile de mai sus se pot precalcuła în timp  $O(N^2)$ ; fiii care conțin în subarbore anumite noduri pot fi precalcułați și ei în  $O(N^2)$ . Așadar rămâne să se găsească intersecția a  $O(N)$  intervale, lucru care se poate face în timp linear folosind un difference array. Ajungem așadar la complexitatea  $O(N^2 + QN)$ .

## C Subtaskul 3

În acest subtask, putem aplica soluția precedentă aproape identic, dar trebuie făcute unele optimizări. În primul rând LCA-urile trebuie calculate în timp logaritm, cu preprocesare  $O(N \log N)$ . Pentru a putea găsi rapid fiul unui nod  $u$  care conține în subarbore un alt nod  $v$ , observăm că acest lucru este echivalent cu a găsi strămoșii lui  $v$  care se găsește la adâncimea cu unu mai mare ca  $u$ , lucru care poate fi calculat în timp logaritm cu preprocesare  $O(N \log N)$ . În ultimul rând, difference array-ul nu va fi menținut pe toate elementele, ci doar pe cele  $O(|S|)$  poziții unde încep sau se termină intervalele a căror intersecție vrem să o calculăm.

## D Subtaskul 4

În acest subtask mulțimea  $S$  doar crește. Echivalent, vrem în continuu să eliminăm din ce în ce mai multe noduri din mulțimea de soluție. Dacă la  $S = \{v_1, \dots, v_k\}$  se adaugă  $v_{k+1}$ , atunci noi trebuie să impunem că în mulțimea de soluție avem că  $\text{dist}(x, v_k) = \text{dist}(x, v_{k+1})$ . Considerând în continuare reprezentarea linearizată de la subtaskul precedent, asta e echivalent cu a elimina din mulțimea de soluție un număr constant de intervale continue de noduri (în ordinea linearizării).

Să sumarizăm ce operații trebuie să le facem. Considerăm numerele  $1, \dots, N$ ; inițial toate sunt considerate a fi “în mulțime”. La o interogare, o subsecvență continuă se elimină din mulțime, și se vrea numărul de elemente rămase încă în mulțime. Vom rezolva aceste interogări cu un arbore de intervale cu propagare lazy cu operația de incrementare/decrementare, și care poate să numere de câte ori apare elementul minim în șir. Un element “în mulțime” va fi reprezentat de un 0 în șirul menținut de arborele de intervale. Când vrem să scoatem o subsecvență din mulțime, o incrementăm. Pentru a găsi numărul de elemente rămase “în mulțime”, vom găsi care este minimul din șir, și de câte ori apare. Dacă minimul este nonzero, atunci sunt 0 elemente rămase “în mulțime”; altfel, sunt la fel de multe elemente ca numărul de apariții a minimului (care trebuie să fie 0).

## E Soluție completă

Soluția este foarte asemănătoare cu soluția de la subtaskul precedent, doar că trebuie să putem elimina elemente din mulțimea  $S$ . Observăm că a impune constrângerea  $\text{dist}(x, v_i) = \text{dist}(x, v_{i+1})$  constă în incrementarea a  $O(1)$  subsecvențe în arborele de intervale menținut; astfel putem “dezimpune” o constrângere *efectiv decrementând* aceleași subsecvențe. Așadar, dacă  $S = \{v_1, \dots, v_k\}$  și se dorește eliminarea lui  $v_i$  (presupunem pentru simplitate că  $1 < i < k$ ; cazul contrar este mai simplu), mai întâi eliminăm constrângerile  $\text{dist}(x, v_{i-1}) = \text{dist}(x, v_i)$  și  $\text{dist}(x, v_i) = \text{dist}(x, v_{i+1})$  (prin decrementarea a  $O(1)$  subsecvențe în arborele de intervale), apoi impunem constrângerea  $\text{dist}(x, v_{i-1}) = \text{dist}(x, v_{i+1})$ . Așadar rezulta complexitatea  $O((N + Q) \log N)$ .