

### Baraj 3 – Seniori

#### Solutie Razbunare

Autori: Razvan Salajan, Adrian Budau, Lucian Bicsi

#### Solutie $O(Q \cdot L)$ - 10

Pentru fiecare query se face dinamica  $dp[i][j]$  = costul minim pentru a ajunge in nodul  $j$  uitandu-ne la primele  $i$  muchii.

#### Solutie $O(L \cdot N^3 + Q \cdot N^2 \cdot \log L)$ - 50

Pentru fiecare element din  $S$  ne tinem  $cost[i][j]$  = costul minim pentru a ajunge din  $i$  in  $j$  folosindu-ne sau nu de muchia din elementul curent. Astfel pentru un query putem folosi un arbore intervale pentru a compune aceste matrici.

#### Solutie $O(L \cdot N \cdot \log L + L \cdot N^2 + Q \cdot (N + \log L))$ - 100

Vom aplica divide et impera. Sa presupunem ca suntem in intervalul  $[st, dr]$  si avem  $mij$  fixat. Vom incerca sa raspundem la toate query-urile care au capatul stanga,  $a \leq mij$  si capatul dreapta,  $mij \leq b$ .

Pentru fiecare interval ne construim 2 dinamici:

- $dpSt[i][j]$  = costul minim pentru a ajunge din  $i$  in  $j$  parcurgand muchiile de la  $mij$  spre stanga;
- $dpDr[i][j]$  = costul minim pentru a ajunge din  $i$  in  $j$  parcurgand muchiile de la  $mij$  spre dreapta;

Cand calculam dinamica  $dpSt[][]$  vom incerca sa si raspundem la toate query-urile care au capatul stanga in pozitia curenta. Cum dinamica  $dpDr[i][j]$  am calculat-o anterior avem 2 optiuni: mai adaugam o stare ( $dpDr[pos][i][j]$  = costul minim de a ajunge din  $i$  in  $j$  daca ne-am uitat la primele  $pos$  muchii) si am obtine complexitatea  $O(L \cdot N^2 \cdot \log L + Q \cdot (N + \log L))$  care ar obtine 75 de puncte, sau ne tinem o matrice  $dpStartDr[i][j]$  = costul minim de a ajunge in nodul  $v$  din al  $i$ -lea query stiind ca am pornit din nodul  $j$ . Pentru a afla raspunsul la al  $i$ -lea query initial cu capatul stang ( $a$ ) in pozitia curenta vom alege costul minim din  $dpSt[x][nod] + dpStartDr[i][nod]$  pentru fiecare  $nod=1, n$ .

Cand calculam dinamica  $dpSt/dpDr[i][j]$  ar trebui actualizate toate perechile de noduri cu costul de refuz (daca se putea ajunge din  $i$  in  $j$  fara muchia curenta). Pentru a evita acest lucru presupunem ca initial refuzam toate muchiile. Asta presupune sa adaugam initial pentru fiecare query suma costurilor de refuz pentru intervalul  $[a, b]$  corespunzator si modificarea pentru fiecare element din  $S$  din  $\langle a, b, c, r \rangle$  in  $\langle a, b, c-r, 0 \rangle$ .

#### Solutie $O(Q \cdot \sqrt{L}) + L \cdot N^2 + L \cdot N \cdot \sqrt{L} + Q \cdot N$ - 90/100 de puncte

O alta abordare este urmatoarea. Se pot rezolva query-urile pe intervale mai mici decat  $\sqrt{L}$  ca la solutia de 10 puncte, in complexitate  $O(\sqrt{L})$ , iar pentru restul query-urilor se putem imparti in  $\sqrt{L}$  clase si sa rezolvam fiecare clasa de query-uri independent. Complexitatea pentru aceasta parte este  $O(Q \cdot \sqrt{L})$

**Baraj 3 – Seniori**

Clasa  $i$  de query-uri va contine toate query-urile care au capatul stanga ( $a$ ) in intervalul  $(\text{sqrt}(L) * i, \text{sqrt}(L) * (i + 1))$ .

Ca si la solutia in complexitate  $O(L * N * \log L + L * N^2 + Q * (N + \log L))$  vom tine doua dinamici, una la stanga si una la dreapta. Pentru cea la stanga se poate adauga o stare si sa se mentine toate matricile  $dpSt[pos][i][j]$  pentru ca sunt cel mult  $\text{sqrt}(L)$  pozitii. Deocamdata complexitatea acestei parti este  $O(L * N^2)$ , iar in partea dreapta nu mai e nevoie sa se tina starea pos, fiindca atunci cand se ajunge la capatul dreapta al unui query, avem informatiile necesare pentru a raspunde la acel query. Complexitatea pentru aceasta parte este  $O(L * N * \text{sqrt}(L) + Q * N)$ .