

S.I. dr. ing. Mugurel Ionut Andreica – Universitatea „Politehnica” București

Soluția 1 – 40 puncte

Se mențin drumurile sub forma de vectori, împreună cu informații asociate fiecărui nod referitor la drumul din care face parte și la poziția sa în drum. Adăugarea unei muchii presupune concatenarea a două drumuri (eventual în sens inversat), iar ștergerea unei muchii presupune separarea unui drum în două drumuri. Cu ajutorul acestor informații menținute, se poate verifica în timp $O(1)$ dacă adăugarea sau ștergerea unei muchii sunt valide, respectiv se poate răspunde în timp $O(1)$ la fiecare interogare (de ex., dacă nodul i se afla pe poziția x în drum, iar drumul este reținut ca un vector, atunci nodurile aflate la distanța D de nodul i se afla pe pozițiile $x-D$, respectiv $x+D$, din cadrul vectorului). Concatenarea a două drumuri și separarea unui drum în două sunt operații ce se execută în timp $O(N)$.

Soluția 2 – 40 puncte

Se menține graful folosind o reprezentare „standard” a grafului (de ex., liste de vecini). Pentru fiecare operație putem parcurge graful (în timp $O(N)$, căci există $O(N)$ noduri și muchii) pentru a decide dacă operația se poate efectua, respectiv pentru a răspunde la interogare.

Soluția 3 – 40 puncte

Se mențin pentru fiecare nod cei maxim 2 vecini din graf. În plus, dacă un nod este capăt de drum, putem memora și capatul opus al aceluiași drum. La adăugarea unei muchii setăm cele două noduri ca fiind vecine unul cu altul. Muchia este adăugată dacă ambele noduri sunt capete de drumuri (au gradul cel mult egal cu 1) și nu sunt capetele aceluiași drum. La ștergerea unei muchii „resetăm” vecinul corespunzător fiecăruia din cele două noduri la „inexistent” (și, eventual, calculăm capatul opus al fiecăruia din cele 2 noduri, care au devenit acum capete de drumuri).

Pentru a răspunde la o interogare vom pleca din nodul i și vom parcurge drumul din care face parte în ambele sensuri, ori până ajungem la capatul său, ori până am parcurs o distanță D . Parcurgerea drumului presupune atenție la implementare, deoarece, în cadrul unui drum, nu putem menține în mod consistent „vecinul stânga”, respectiv „vecinul dreapta” al fiecărui nod (decat dacă efectuăm procesări suplimentare la adăugarea unei muchii, adică la concatenarea a două drumuri). Din fericire, menținerea consistentă a vecinilor „stânga” și „dreapta” în cadrul unui drum nu este necesară. Când parcurgem drumul într-un sens, menținem atât nodul curent, cât și nodul anterior. Apoi, următorul nod ce va fi parcurs este acel vecin al nodului curent care este diferit de cel anterior.

Această soluție are complexitatea $O(N)$ pentru fiecare interogare și $O(1)$ pentru un tip de modificare a grafului și $O(N)$ pentru celălalt tip de modificare a grafului.

Soluția 4 – 100 puncte

Vom extinde soluția anterioară pentru a menține, pentru fiecare nod, pe lângă cei maxim 2 vecini, încă maxim 2 **K-vecini**, adică cele maxim 2 noduri aflate la distanță exact K de nodul respectiv. Dacă notăm cei doi vecini ai nodului i prin $v1[i]$ și $v2[i]$ și cei doi K-vecini ai nodului prin $kv1[i]$ și $kv2[i]$, vom avea proprietatea că $kv1[i]$ este K-vecinul nodului i din același sens ca și vecinul $v1[i]$ (și similar pentru $kv2[i]$ și $v2[i]$).

La adăugarea unei muchii maxim K noduri de fiecare parte a muchiei vor obține K-vecini noi în sensul dinspre muchia adăugată. La ștergerea unei muchii, maxim K noduri de fiecare parte a muchiei își vor pierde K-vecinii din sensul înspre muchia stearsă.

Folosind K-vecinii, putem răspunde la interogări în timp $O(K+N/K)$. Mai întâi vom „sări” din K , până când un nou salt de lungime K ar „sări” prea departe (ori peste distanța D , ori în afara drumului). Apoi vom continua să ne deplasăm din 1 în 1 (din vecin în vecin) până ajungem la distanța D sau la capatul drumului. Și acest mod de parcurgere necesită aceeași atenție la implementare din soluția anterioară (întrucât nu menținem în mod consistent vecini și K-vecini „stânga” și „dreapta”, următorul vecin/K-vecin ce va fi parcurs este acel vecin/K-vecin al nodului curent care este diferit de vecinul/K-vecinul parcurs anterior).

Dacă alegem $K=O(\sqrt{N})$, atunci complexitatea de timp a fiecărei operații este $O(\sqrt{N})$.