



Vilkatis

Japānas prefektūrā Ibaraki ir N pilsētas un M ceļi. Pilsētas ir numurētas no 0 līdz $N - 1$ iedzīvotāju skaita pieaugšanas secībā. Katrs ceļš savieno divas dažādas pilsētas, un var tikt izmantots abos virzienos. Jūs varat nokļūt no jebkuras pilsētas uz jebkuru citu pilsētu izmantojot vienu vai vairākus ceļus.

Jūs esat ieplānojuši Q ceļojumus, kas sanumurēti no 0 līdz $Q - 1$. Ceļojums ar numuru i ($0 \leq i \leq Q - 1$) sākas pilsētā ar numuru S_i un beidzas pilsētā ar numuru E_i .

Jūs esat vilkatis. Jums ir divas formas: **cilvēka** un **vilka**. Katra ceļojuma sākumā jūs esat cilvēka formā. Katra ceļojuma beigās jums ir jābūt vilka formā. Ceļojuma laikā jums ir **jātransformējas** (jāmainās no cilvēka formas uz vilka formu) tieši vienreiz. Jūs varat transformēties tikai kādā no pilsētām (varbūt arī S_i vai E_i).

Vilkača dzīve nav viegla. Jums ir jāaizvairās no pilsētām ar nelielu iedzīvotāju skaitu, kad jūs esat cilvēka formā, un ir jāaizvairās no pilsētām ar lielu iedzīvotāju skaitu, kad jūs esat vilka formā. Katram ceļojumam ar numuru i ($0 \leq i \leq Q - 1$), ir divas robežas L_i un R_i ($0 \leq L_i \leq R_i \leq N - 1$) kas raksturo no kurām pilsētām ir jāaizvairās. Precīzāk, jums ir jāaizvairās no pilsētām ar numuriem $0, 1, \dots, L_i - 1$ kad jūs esat cilvēka formā, un ir jāaizvairās no pilsētām ar numuriem $R_i + 1, R_i + 2, \dots, N - 1$ kad jūs esat vilka formā. Tas nozīmē ka ceļojumā ar numuru i , jūs varat transformēties tikai vienā no pilsētām ar numuriem $L_i, L_i + 1, \dots, R_i$.

Jūsu uzdevums ir katram ceļojumam noskaidrot vai ir iespējams aizceļot no pilsētas ar numuru S_i līdz pilsētai ar numuru E_i tā, lai visi augstākminētie ierobežojumi izpildītos. Jūsu izveidotajam maršrutam var būt jebkurš garums.

Implementācijas detaļas

Jums ir jāimplementē šāda funkcija:

```
int[] check_validity(int N, int[] X, int[] Y, int[] S, int[] E, int[] L, int[] R)
```

- N : pilsētu skaits.
- X un Y : masīvi garumā M . Katram j ($0 \leq j \leq M - 1$), ceļš savieno pilsētas ar numuriem $X[j]$ un $Y[j]$.
- S , E , L , un R : masīvi garumā Q , kas apraksta ceļojumus.

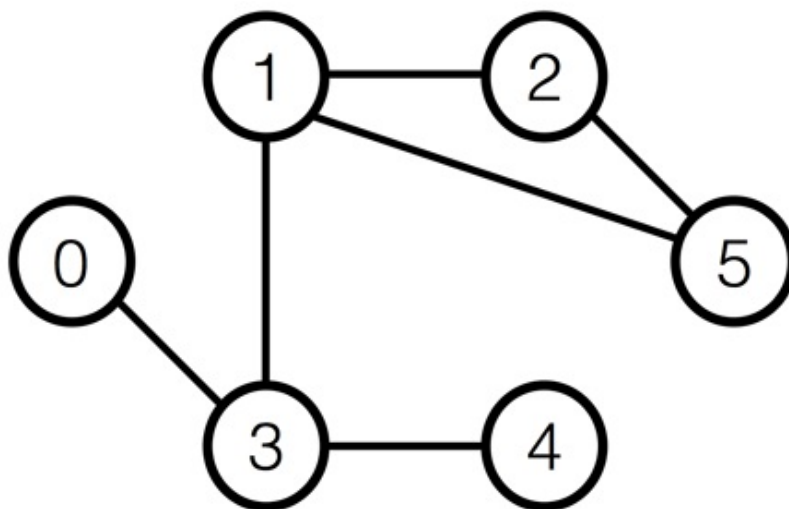
Pievērsiet uzmanību, ka vērtības M and Q ir masīvu garumi, un var tikt iegūti kā norādīts implementācijas norādījumos.

Funkcija `check_validity` tiks izsaukta tieši vienu reizi katram testam. Šai funkcijai ir jāatgriež veselu skaitļu masīvs A garumā Q . A_i ($0 \leq i \leq Q - 1$) vērtībai ir jābūt 1, ja ceļojums ar numuru i ir iespējams izpildot visus augstākminētos ierobežojumus, vai 0 pretējā gadījumā.

Piemērs

Pieņemsim, ka $N = 6$, $M = 6$, $Q = 3$, $X = [5, 1, 1, 3, 3, 5]$, $Y = [1, 2, 3, 4, 0, 2]$, $S = [4, 4, 5]$, $E = [2, 2, 4]$, $L = [1, 2, 3]$, un $R = [2, 2, 4]$.

Vērtētājs izsauc `check_validity(6, [5, 1, 1, 3, 3, 5], [1, 2, 3, 4, 0, 2], [4, 4, 5], [2, 2, 4], [1, 2, 3], [2, 2, 4])`.



Ceļojumam ar numuru 0, jūs varat aizceļot no pilsētas ar numuru 4 līdz pilsētai ar numuru 2 šādi:

- Sākat ceļu pilsētā ar numuru 4 (Jūs esat cilvēka formā)
- Aiziet uz pilsētu ar numuru 3 (Jūs esat cilvēka formā)
- Aiziet uz pilsētu ar numuru 1 (Jūs esat cilvēka formā)
- Transformēties vilka formā (Jūs esat vilka formā)
- Aiziet uz pilsētu ar numuru 2 (Jūs esat vilka formā)

Ceļojumiem ar numuriem 1 un 2, jūs nevarat aizceļot no vienas dotās pilsētas uz otru.

Tātad, jūsu funkcijai ir jāatgriež `[1, 0, 0]`.

Faili `sample-01-in.txt` un `sample-01-out.txt` uzdevuma arhīvā atbilst šim piemēram. Arhīvā ir pieejami arī citu piemēru ievaddati un izvaddati.

Ierobežojumi

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$
- Katram $0 \leq j \leq M - 1$
 - $0 \leq X_j \leq N - 1$
 - $0 \leq Y_j \leq N - 1$
 - $X_j \neq Y_j$
- Jums ir jāceļo no jebkuras pilsētas uz jebkuru citu pilsētu izmantojot tikai ceļus.
- Katru pilsētu pāri tieši savieno ne vairāk kā viens ceļš. Citiem vārdiem, visiem $0 \leq j < k \leq M - 1$, $(X_j, Y_j) \neq (X_k, Y_k)$ un $(Y_j, X_j) \neq (X_k, Y_k)$.
- Katram $0 \leq i \leq Q - 1$
 - $0 \leq L_i \leq S_i \leq N - 1$
 - $0 \leq E_i \leq R_i \leq N - 1$
 - $S_i \neq E_i$
 - $L_i \leq R_i$

Apakšuzdevumi

1. (7 punkti) $N \leq 100$, $M \leq 200$, $Q \leq 100$
2. (8 punkti) $N \leq 3\,000$, $M \leq 6\,000$, $Q \leq 3\,000$
3. (34 punkti) $M = N - 1$ un katra pilsēta ir pa tieši savienota ar ne vairāk kā 2 ceļiem (pilsētas ir savienotas līnijā)
4. (51 punkts) Bez papildu ierobežojumiem.

Paraugvērtētājs

Paraugvērtētājs lasa ievaddatus šādā formātā:

- 1. rinda: $N M Q$
- $2 + j$ -tā rinda ($0 \leq j \leq M - 1$): $X_j Y_j$
- $2 + M + i$ -tā rinda ($0 \leq i \leq Q - 1$): $S_i E_i L_i R_i$

Paraugvērtētājs izvada funkcijas `check_validity` atgriežamo vērtību šādi:

- $1 + i$ -tā rinda ($0 \leq i \leq Q - 1$): A_i