



მგელკაცა

იაპონიის იბარაქის პრეფექტურაში გვაქვს N ქალაქი და M გზა. ქალაქები დანომრილია 0 -დან $(N - 1)$ -ის ჩათვლით მოსახლეობის ზრდის მიხედვით. ყოველი გზა აკავშირებს ქალაქების სხვადასხვა წყვილს და მასზე შესაძლებელია მოძრაობა ორმხრივად. ამ გზების გამოყენებით თქვენ შეგიძლიათ იმოგზაუროთ ნებისმიერი ქალაქიდან ნებისმიერ სხვა ქალაქში ამ გზებიდან ერთი ან რამდენიმე გზის გამოყენებით.

თქვენ დაგეგმეთ Q რაოდენობის მოგზაურობა, დანომრილი 0 -დან $(Q - 1)$ -ის ჩათვლით. i -ური მოგზაურობა ($0 \leq i \leq Q - 1$) ეს არის მოგზაურობა S_i ქალაქიდან E_i ქალაქში.

თქვენ ხართ მგელკაცა, ანუ გაქვთ ორი ფორმა: **ადამიანის ფორმა** და **მგლის ფორმა**. ყოველი მოგზაურობის დაწყებისას თქვენ იმყოფებით ადამიანის ფორმაში. ყოველი მოგზაურობის დამთავრებისას თქვენ უნდა იყოთ მგლის ფორმაში. მოგზაურობის განმავლობაში თქვენ უნდა **გარდაიქმნათ** (შესვალთ ადამიანის ფორმა მგლის ფორმით) მხოლოდ ერთხელ და ეს უნდა მოხდეს, როცა თქვენ იმყოფებით რომელიმე ქალაქში (შესაძლებელია S_i ან E_i -ც).

მგელკაცას ცხოვრება ადვილი არ არის. თქვენი გამოცდილებით იცით, რომ როცა იმყოფებით ადამიანის ფორმაში უმჯობესია გვერდი აუაროთ უფრო მცირედ დასახლებულ ქალაქებს, ხოლო როცა იმყოფებით მგლის ფორმაში, გვერდი უნდა აუაროთ უფრო მრავალრიცხოვან ქალაქებს. კონკრეტულად, ყოველი i -ური მოგზაურობისას, თქვენ ირჩევთ ორ მთელ L_i და R_i რიცხვს, სადაც $0 \leq L_i \leq R_i \leq N - 1$. i -ური მოგზაურობის დროს ადამიანის ფორმაში თქვენ გადაწყვიტეთ გვერდი აუაროთ ქალაქებს $0, 1, \dots, L_i - 1$, ხოლო მგლის ფორმაში გვერდი აუაროთ $R_i + 1, R_i + 2, \dots, N - 1$ ქალაქებს. კერძოდ ეს ნიშნავს, რომ თქვენ უნდა გარდაიქმნათ ერთ-ერთ $L_i, L_i + 1, \dots, R_i$ ქალაქში.

ყოველი მოგზაურობისას თქვენი ამოცანაა განსაზღვროთ შესაძლებელია თუ არა მოგზაურობა S_i ქალაქიდან E_i ქალაქში, ისე რომ დააკმაყოფილოთ ზემოთ აღწერილი პირობები. მარშრუტს, რომელსაც თქვენ ირჩევთ, შეიძლება ჰქონდეს ნებისმიერი სიგრძე.

იმპლემენტაციის დეტალები

თქვენ იმპლემენტაცია უნდა გაუკეთოთ შემდეგ ფუნქციას:

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

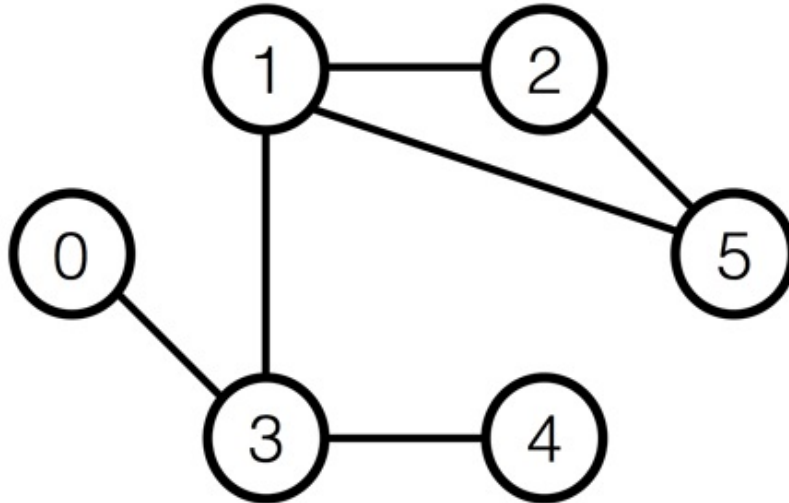
- N : ქალაქების რაოდენობა.
- X და Y : M სიგრძის მასივები. ყოველი j -თვის ($0 \leq j \leq M - 1$), ქალაქი $X[j]$ უშუალოდ უკავშირდება გზით $Y[j]$ ქალაქს.
- S , E , L , და R : Q სიგრძის მასივები წარმოდგენილი მოგზაურობისთვის.

ფუნქცია `check_validity` გამოიძახება ზუსტად ერთხელ ყოველი ტესტისათვის. ამ ფუნქციამ უნდა დაგვიბრუნოს მთელი რიცხვების Q სიგრძის A მასივი. A_i -ის მნიშვნელობა ($0 \leq i \leq Q - 1$) უნდა იყოს 1, თუ შესაძლებელია მოგზაურობა S_i ქალაქიდან E_i ქალაქში, როცა გვერდს აუვლით $0, 1, \dots, L_i - 1$ ქალაქებს ადამიანის ფორმაში და $R_i + 1, R_i + 2, \dots, N - 1$ ქალაქებს მგლის ფორმაში. წინააღმდეგ შემთხვევაში უნდა იყოს 0.

მაგალითი

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

გრაფის გამოძახება `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])`.



0 ნომრის მოგზაურობისას, თქვენ შეგიძლიათ 4 ქალაქიდან 2 ქალაქში შემდეგნაირად:

- სტარტს ავიღებთ 4 ქალაქიდან (თქვენ გაქვთ ადამიანის ფორმა)
- ჩახვალთ 3 ქალაქში (თქვენ გაქვთ ადამიანის ფორმა)
- შემდეგ ჩახვალთ 1 ქალაქში (თქვენ გაქვთ ადამიანის ფორმა)
- გარდაიქმნებით მგლის ფორმაში (თქვენ გაქვთ მგლის ფორმა)
- ჩახვალთ 2 ქალაქში (თქვენ გაქვთ მგლის ფორმა)

1 და 2 მოგზაურობისას თქვენ არ შეგიძლიათ მოგზაურობა მოცემულ ქალაქებს შორის. შედეგად თქვენმა პროგრამამ უნდა დააბრუნოს $[1, 0, 0]$.

მიბმულ დაარქივებულ პაკეტში არსებული sample-01-in.txt და sample-01-out.txt ფაილები შეესაბამება მოცემულ მაგალითს. შეტანა/გამოტანის სხვა მაგალითებიც ამავე პაკეტშია მოცემული.

შეზღუდვები

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$
- $0 \leq X_j \leq N - 1$ ($0 \leq j \leq M - 1$)
- $0 \leq Y_j \leq N - 1$ ($0 \leq j \leq M - 1$)
- თქვენ შეგიძლიათ იმოგზაუროთ ნებისმიერი ქალაქიდან ნებისმიერ სხვა ქალაქამდე გზების გამოყენებით.
- ქალაქთა ყოველი წყვილი დაკავშირებულია უშუალოდ არა უმეტეს ერთი გზით. სხვა სიტყვებით, ყველა $(0 \leq j < k \leq M - 1)$ -თვის, $(X_j, Y_j) \neq (X_k, Y_k)$ და $(Y_j, X_j) \neq (X_k, Y_k)$.
- ყველა $0 \leq i \leq Q - 1$ -სათვის
 - $0 \leq L_i \leq S_i \leq N - 1$
 - $0 \leq E_i \leq R_i \leq N - 1$
 - $S_i \neq E_i$ ($0 \leq i \leq Q - 1$) - $0 \leq L_i \leq R_i \leq N - 1$

ქვეამოცანები

1. (7 ქულა) $N \leq 100$, $M \leq 200$, $Q \leq 100$
2. (8 ქულა) $N \leq 3\,000$, $M \leq 6\,000$, $Q \leq 3\,000$
3. (34 ქულა) $M = N - 1$ და არცერთი ქალაქი არ არის დაკავშირებული 2-ზე მეტ ქალაქთან (ქალაქები დაკავშირებული არიან ერთ წრფივ ხაზზე)
4. (51 ქულა) დამატებითი შეზღუდვების გარეშე.

გრაფერის მაგალითი

სანიმუშო გრაფერი კითხულობს შესატან მონაცემებს შემდეგი ფორმატით:

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

სანიმუშო გრაფერი ბეჭდავს ფუნქცია check_validity-ის მიერ დაბრუნებულ მნიშვნელობას შემდეგი ფორმატით:

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