



Kurtadam

Japonya Ibaraki bölgesinde N tane şehir ve M tane yol vardır. Şehirler 0 'dan $N - 1$ 'e artan nüfusa göre sıralanmışlardır. Her bir yol birbirinden farklı bir şehir çiftini bağlar ve yolda her iki yönde gidilebilir. Bir şehirden diğerine bu yolların bir ya da daha fazlasını kullanarak gidebilirsiniz.

0 'dan $Q - 1$ 'e numaralandırılmış Q tane yolculuk planladınız. i . yolculukta ($0 \leq i \leq Q - 1$) S_i şehrinden E_i şehrine yolculuk etmeyi hedeflemektesiniz.

Siz bir kurtadamsınız. İki şekle bürünebiliyorsunuz: **insan şekli** ve **kurt şekli**. Her yolculuğun başında insan şeklindesiniz. Yolculuğun sonunda, kurt şeklinde olmanız gerekmektedir. Yani yolculuk sırasında tam olarak bir kez **şekil değiştirmelisiniz** (insan şeklinden kurt şekline geçmelisiniz). Yalnızca, bir şehirde bulunduğunuz zaman şekil değiştirebilirsiniz (bu şehir S_i ya da E_i de olabilir).

Bir kurtadam olarak yaşamak hiç de kolay değildir. İnsan şeklinde olduğunuz zaman az-nüfuslu şehirlere gitmemeli ve kurt şeklinde olduğunuz zamanlarda da yüksek-nüfuslu şehirlere gitmemelisiniz. Her bir i ($0 \leq i \leq Q - 1$) yolculuğu için L_i ve R_i ($0 \leq L_i \leq R_i \leq N - 1$) diye adlandırılmış iki eşik değeri bulunmaktadır. Bu eşik değerleri hangi şehirlere hangi şekilde gidilip gidilemeyeceğini göstermektedir. Detaylı açıklamak gerekirse, insan şeklinde olduğunuz zamanlarda $0, 1, \dots, L_i - 1$ şehirlerine gitmemeli ve kurt şeklinde olduğunuz zamanlarda da $R_i + 1, R_i + 2, \dots, N - 1$ şehirlerine gitmemelisiniz. Bunun anlamı i yolculuğunda sadece $L_i, L_i + 1, \dots, R_i$ şehirlerinden yalnızca birisinde şekil değiştirebileceğinizdir.

Göreviniz, her bir yolculuk için, S_i şehrinden E_i şehrine yukarıda tanımlanan kısıtlar doğrultusunda gitmenin mümkün olup olmadığını belirlemektir.

İki şehir arasındaki yolculuk rotası herhangi bir uzunlukta olabilir.

Kodlama detayları

Aşağıdaki fonksiyonu yazmanız gerekmektedir:

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

- N : şehir sayısı.
- X ve Y : M uzunluğunda diziler. Her bir j ($0 \leq j \leq M - 1$) için, $X[j]$ şehri $Y[j]$

şehrine bir yol ile direkt bağlıdır.

- S , E , L , ve R : Q uzunluğunda diziler, yolculuk bilgilerini ve kısıtları belirtmektedir.

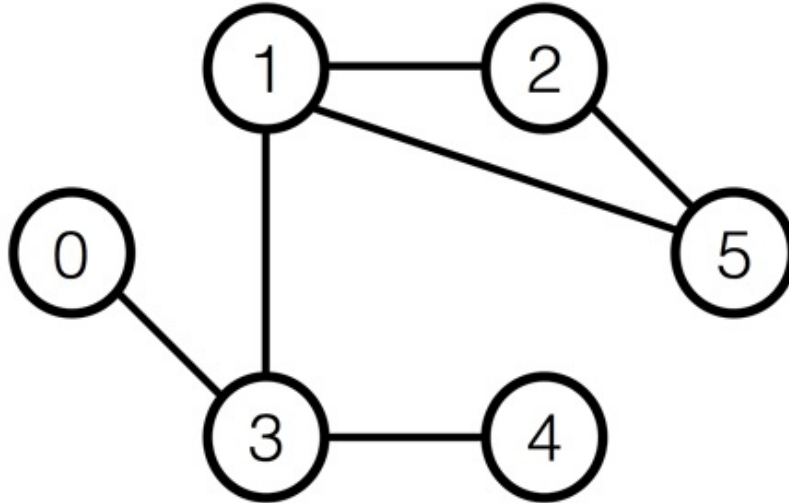
M ve Q değerleri dizilerin uzunluğunu göstermektedir ve programlama duyurusu dokümanında belirtildiği gibi elde edilebilirler.

`check_validity` fonksiyonu her bir test case için tam olarak bir kez çağrılır. Bu fonksiyon Q uzunluğunda tam sayılardan oluşan bir A dizisi dönmelidir. A_i ($0 \leq i \leq Q - 1$) değeri, eğer i . yolculuk yukarıdaki kısıtlar dahilinde mümkünse, 1 olmalı, mümkün değilse 0 olmalıdır.

Örnek

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

Değerlendirici `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])` çağırır.



0'ıncı yolculuk için, 4'üncü şehirden 2'inci şehire aşağıdaki şekilde gidebilirsiniz:

- 4. şehirden başla (İnsan seklindesiniz)
- 3. şehire git (İnsan seklindesiniz)
- 1. şehire git (İnsan seklindesiniz)
- Kendini kurt şekline dönüştür (Kurt seklindesiniz)
- 2. şehire git (Kurt seklindesiniz)

1'inci ve 2'inci yolculuklar için, verilen şehirler arasında yolculuk edemezsiniz.

Dolayısıyla, programınız `[1, 0, 0]` dönmelidir.

Ekteki ziplenmiş paketteki `sample-01-in.txt` ve `sample-01-out.txt` dosyaları bu örneğe karşılık gelmektedir. Pakette bundan başka girdi/çıkı örnekleri de bulunmaktadır.

Kısıtlar

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$
- For each $0 \leq j \leq M - 1$
 - $0 \leq X_j \leq N - 1$
 - $0 \leq Y_j \leq N - 1$
 - $X_j \neq Y_j$
- Herhangi bir şehirden diğer herhangi bir şehre yolları kullanarak gidebilirsiniz.
- Herhangi bir şehir çifti, direkt olarak en fazla bir yolla bağlıdır. Başka bir ifadeyle, her $0 \leq j < k \leq M - 1$ için, $(X_j, Y_j) \neq (X_k, Y_k)$ ve $(Y_j, X_j) \neq (X_k, Y_k)$.
- Her bir $0 \leq i \leq Q - 1$ için,
 - $0 \leq L_i \leq S_i$
 - $0 \leq E_i \leq R_i \leq N - 1$
 - $S_i \neq E_i$ ($0 \leq i \leq Q - 1$)
 - $0 \leq L_i \leq R_i \leq N - 1$

Altgörevler

1. (7 puan) $N \leq 100$, $M \leq 200$, $Q \leq 100$
2. (8 puan) $N \leq 3\,000$, $M \leq 6\,000$, $Q \leq 3\,000$
3. (34 puan) $M = N - 1$ ve her şehir en fazla 2 yola bağlıdır (şehirler bir doğru üzerinde bağlıdır.)
4. (51 puan) Herhangi bir ek kısıt yok

Örnek değerlendirici

Örnek değerlendirici girdiyi aşağıdaki formatta okur:

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

Örnek değerlendirici, `check_validity` 'nin döndüğü değeri aşağıdaki formatta yazar:

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