

World directors (directors)

Linus ve Tinus kısa süre önce dünyanın yeni iki *Direktörleri* olarak atandılar.

Dünya, N şehir ve $N - 1$ yoldan oluşuyor ve herhangi iki şehir arasında belirli bir yol dizisi kullanarak hareket etmek mümkün. İki şehir arasındaki mesafe, bir şehirden diğerine gitmek için geçilmesi gereken minimum yol sayısıdır.

Yeni atanan direktörler olarak Linus ve Tinus'un, tüm dünyayı kapsayan geleneksel bir *devriye* gerçekleştirmesi bekleniyor. Bu devriye şunlardan oluşur:

- İlk olarak, Linus ve Tinus belirli başlangıç şehirleri olan X ve Y 'ye gider.
- Daha sonra, her gün iki direktörden biri, mevcut şehirden tek bir yolla ulaşılabilen bir şehre hareket eder. Bu, her iki direktör de her şehirde en az bir kez bulunana ve başlangıç şehirlerine dönene kadar devam eder. Bir direktörün art arda birden fazla gün hareket edebileceğini unutmayın. Yani iki direktörün sırayla hareket etmesi gerekmez.

Linus ve Tinus, devriyelerinin, birbirlerinden ne kadar uzak olurlarsa o kadar ciddiye kabul edileceğini çok iyi biliyorlar: bir devriyenin *ciddiyeti*, birbirlerine en yakın oldukları zamanki iki direktör arasındaki mesafedir.

Bu nedenle iki direktör, devriyelerini planlamalarına yardımcı olmanız için sizi işe aldı ve sizin göreviniz, aşağıdaki türden Q soruyu yanıtlamak:

- Linus X şehirden ve Tinus Y şehirden başlarsa, devriyeleriyle ulaşabilecekleri maksimum ciddiyet değeri nedir?

Implementasyon

.cpp uzantılı tek bir dosya göndermelisiniz.



Bu görev için ekler arasında, örnek bir uygulama içeren `directors.cpp` şablonu bulacaksınız.

Aşağıdaki fonksiyonları kodlamanız gerekiyor:

C++

```
void init(int N, vector<int> A, vector<int> B);
```

- N tam sayısı, şehir sayısıdır.
- A ve B dizileri, 0 'dan $N - 2$ 'ye kadar indekslenmiş olup yolları içerir. Özellikle, i -inci yol A_i ve B_i şehirlerini birbirine bağlar.
- Bu fonksiyon, programınızın çalıştırılmasının başında bir kez çağrılacaktır.

C++

```
int patrol(int X, int Y);
```

- X ve Y tam sayıları, Linus ve Tinus'un başlangıç şehirleridir.
- Bu fonksiyon, Linus ve Tinus'un X ve Y şehirlerinden başladığı bir devriyenin ulaşabileceği maksimum ciddiyet değerini dönmelidir.
- Bu fonksiyon, programınızın çalıştırılması sırasında Q kez çağrılacaktır.

Örnek Değerlendirici

Düzeltilme sırasında kullanılan değerlendiricinin (grader) basitleştirilmiş bir sürümü, bu problemle ilgili dizinde mevcuttur. Çözümlerinizi yerel olarak test etmek için kullanabilirsiniz. Örnek değerlendirici (grader), `stdın` dosyasından girdi verilerini okur, uygulamanız gereken fonksiyonu çağırır ve aşağıdaki formatta `stdout` dosyasına yazar.

Girdi dosyası $N + Q$ satırdan oluşur ve şunları içerir:

- Satır 1: N ve Q tam sayıları.
- Satır $2 + i$ ($0 \leq i < N - 1$): A_i ve B_i tam sayıları.
- Satır $N + 1 + j$ ($0 \leq j < Q$): X_j ve Y_j tam sayıları.

Çıktı dosyası Q satırdan oluşur ve `patrol` fonksiyonunun döndüğü değerleri içerir.

Kısıtlamalar

- $1 \leq N \leq 200\,000$.
- $1 \leq Q \leq 100\,000$.
- Her sorguda $0 \leq X, Y < N$.

Puanlama

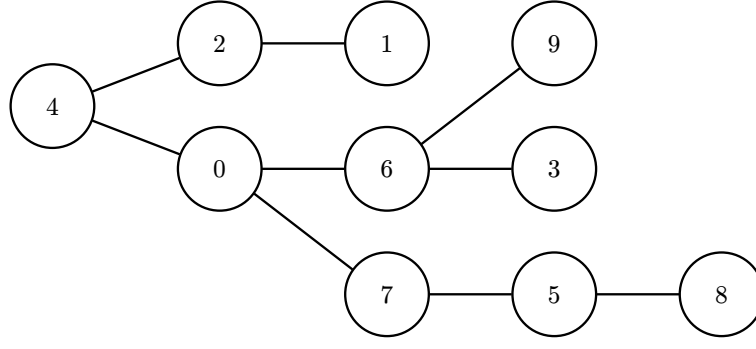
- **Alt görev 0 [0 puan]:** Örnek.
- **Alt görev 1 [8 puan]:** Tüm $0 \leq i < N - 1$ için $A_i = 0$, $B_i = i + 1$.
- **Alt görev 2 [16 puan]:** Tüm $0 \leq i < N - 1$ için $A_i = 0$, $B_i = i + 1$ veya $A_i = i$, $B_i = i + 1$.
- **Alt görev 3 [13 puan]:** $N, Q \leq 200$.
- **Alt görev 4 [14 puan]:** $N \leq 1000$.
- **Alt görev 5 [18 puan]:** Her sorguda, X ve Y tüm başlangıç şehirleri arasında cevabı maksimize eder.
- **Alt görev 6 [17 puan]:** $Q \leq 200$.
- **Alt görev 7 [14 puan]:** Ek kısıtlama yok.

Örnekler

stdın	stdout
10 3	2
0 4	1
1 2	2
8 5	
6 0	
9 6	
2 4	
7 0	
3 6	
5 7	
9 8	
0 6	
6 4	

Açıklama

İlk örnek durumda dünya aşağıdaki yapıdadır:



İlk sorgu için, aşağıdaki hareketlerle 2 ciddiye bir devriyeye ulaşabiliriz:

- Linus şu yolu izler: $9 \rightarrow 6 \rightarrow 3 \rightarrow 6 \rightarrow 0 \rightarrow 4 \rightarrow 2 \rightarrow 1$;
- Tinus şu yolu izler: $8 \rightarrow 5 \rightarrow 7 \rightarrow 0 \rightarrow 6 \rightarrow 3 \rightarrow 6 \rightarrow 9$;
- Linus şu yolu izler: $1 \rightarrow 2 \rightarrow 4 \rightarrow 0 \rightarrow 7 \rightarrow 5 \rightarrow 8$;
- Tinus şu yolu izler: $9 \rightarrow 6 \rightarrow 3 \rightarrow 6 \rightarrow 0 \rightarrow 4 \rightarrow 2 \rightarrow 1$;
- Linus şu yolu izler: $8 \rightarrow 5 \rightarrow 7 \rightarrow 0 \rightarrow 6 \rightarrow 3 \rightarrow 6 \rightarrow 9$;
- Tinus şu yolu izler: $1 \rightarrow 2 \rightarrow 4 \rightarrow 0 \rightarrow 7 \rightarrow 5 \rightarrow 8$.

Ciddiyet değerinin 3 veya daha fazla olduğu bir devriyenin bulunmadığı gösterilebilir.