



Binlerce Ada

Binlerce Ada, Java Denizi'nde bulunan bir grup adadır. Bu grup, 0 ile $N - 1$ arasında numaralandırılmış N tane adadan oluşur.

Adalar arasında yelken açmak için kullanılacak, 0 ile $M - 1$ arasında numaralandırılmış M tane kano vardır. $0 \leq i \leq M - 1$ olacak şekilde her i için, kano i , $U[i]$ veya $V[i]$ adalarından birine demirlenebilir ve sadece $U[i]$ ve $V[i]$ adaları arasında yelken açmak için kullanılabilir. Daha açık olarak, kano $U[i]$ adasına demirlendiğinde, bu kano $U[i]$ adasından $V[i]$ adasına yelken açmak için kullanılabilir, ve ardından kano $V[i]$ adasına demirlenir. Benzer şekilde, kano $V[i]$ adasına demirlendiğinde, bu kano $V[i]$ adasından $U[i]$ adasına yelken açmak için kullanılabilir, ve ardından kano $U[i]$ adasına demirlenir. Başlangıçta, kano $U[i]$ adasına demirlenmiştir. Aynı ada çifti arasında yelken açmak için birden fazla kano kullanılması mümkündür. Aynı adaya birden fazla kanonun demirlenmesi de mümkündür.

Güvenlik nedeniyle, bir kanoya her seferden sonra bakım yapılması gerekir, bu da aynı kanoyla arka arkaya iki kez yelken açılmasını yasaklar. Yani, kano i kullanıldıktan sonra, başka bir kano, kano i tekrar kullanılmadan önce kullanılmalıdır.

BuDengklek isimindeki kişi, bazı adalar arasında bir yolculuk planlamak istiyor. Yolculuğu ancak ve ancak aşağıdaki koşullar yerine getirildiğinde **geçerlidir**.

- Yolculuğuna 0 adasında başlar ve yolculuğu 0 adasında biter.
- Ada 0 dışında en az bir adayı ziyaret eder.
- Yolculuk sona erdiğinde her kano, yolculuk öncesindeki adasında demirlenir. Yani, $0 \leq i \leq M - 1$ olacak şekilde her i için kano i , $U[i]$ adasına demirlenmelidir.

Sizden BuDengklek'in en fazla 2 000 000 kez yelken açacak şekilde geçerli bir yolculuk bulmasına yardım etmeniz veya böyle bir geçerli yolculuk yoksa bunu belirlemeniz istenmektedir. Bu görevde belirtilen kısıtlamalar altında (Kısıtlar bölümüne bakın), geçerli bir yolculuk varsa, aynı zamanda 2 000 000'dan fazla seferi içermeyen geçerli bir yolculuğun da olduğu ispatlanabilir.

Programlama Detayları

Aşağıdaki fonksiyonu kodlamalısınız:

```
union(bool, int[]) find_journey(int N, int M, int[] U, int[] V)
```

- N : ada sayısı.
- M : kano sayısı.
- U, V : kanoları tanımlayan M uzunluğundaki diziler.
- Bu fonksiyon ya bir boolean ya da bir tamsayı dizisi return etmelidir:
 - Geçerli bir yolculuk yoksa fonksiyon false return etmelidir.
 - Geçerli bir yolculuk varsa fonksiyon, geçerli yolculuğu temsil eden en fazla 2 000 000 tamsayıdan oluşan bir dizi return etmelidir. Tam puan alabilmek için bu dizinin elemanları yolculukta kullanılan kanoların numaralarını (kullanım sırasına göre) vermelidir. Aksi takdirde, eğer fonksiyon true, 2 000 000'dan fazla tamsayı elemanı olan bir dizi veya geçerli bir yolculuğu tanımlamayan bir tamsayı dizisi return ederse, çözüme kısmi bir puan verilir (Alt Görevler bölümüne bakın).
- Bu fonksiyon tam olarak bir kere çağrılır.

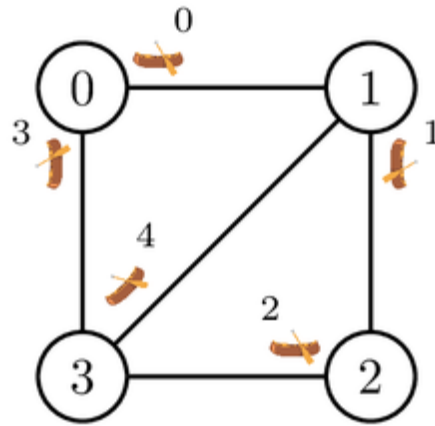
Örnekler

Örnek 1

Aşağıdaki çağırımı göz önüne alın:

```
find_journey(4, 5, [0, 1, 2, 0, 3], [1, 2, 3, 3, 1])
```

Adalar ve kanolar aşağıdaki resimde görülmektedir.



Olası bir geçerli yolculuk aşağıdaki gibidir. BuDengkek ilk olarak sırasıyla 0, 1, 2 ve 4 kanoları ile yelken açar. Bunun sonucunda, BuDengkek 1 adasındadır. Bundan sonra, BuDengkek şu anda 1 adasına demirlendiğinden ve kullandığı son kano, kano 0 olmadığı için kano 0 ile tekrar yelken açabilir. Kano 0 ile tekrar yelken açtıktan sonra, BuDengkek şimdi 0 adasındadır. Ancak, son durumda 1, 2 ve 4 kanoları yolculuktan önceki adalarına demirlenmemiştir. BuDengkek daha sonra yine 3, 2, 1, 4 ve 3 kanoları ile yelken açarak yolculuğuna devam eder. BuDengkek 0 adasına geri döner ve tüm kanolar yolculuktan önceki adalara demirlenmiştir.

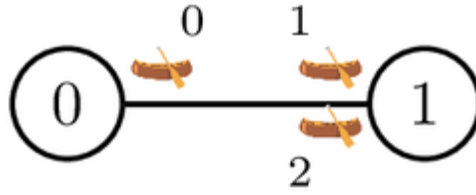
Bu nedenle $[0, 1, 2, 4, 0, 3, 2, 1, 4, 3]$ fonksiyonun geçerli bir return değeridir.

Örnek 2

Aşağıdaki çağırımı göz önüne alın:

```
find_journey(2, 3, [0, 1, 1], [1, 0, 0])
```

Adalar ve kanolar aşağıdaki resimde görülmektedir.



BuDenklek sadece 0 kanosu ile yelken açabilir, ardından ya kano 1 ya da kano 2 ile yelken açabilir. BuDenklek'in arka arkaya iki kez kano 0 ile yelken açamayacağını unutmayın. Her iki durumda da, BuDenklek 0 adasına geri döner. Ancak, kanolar yolculuktan önceki adalarda değildir ve 0 adasına demirlenen tek kano az önce kullanılan kano olduğundan, BuDenklek daha sonra herhangi bir kano ile yelken açamaz. Geçerli bir yolculuk olmadığı için fonksiyon false return etmelidir.

Kısıtlar

- $2 \leq N \leq 100\,000$
- $1 \leq M \leq 200\,000$
- $0 \leq U[i] \leq N - 1$ ve $0 \leq V[i] \leq N - 1$ ($0 \leq i \leq M - 1$ olacak şekilde her i için)
- $U[i] \neq V[i]$ ($0 \leq i \leq M - 1$ olacak şekilde her i için)

Alt görevler

1. (5 puan) $N = 2$
2. (5 puan) $N \leq 400$. $0 \leq x < y \leq N - 1$ şartını sağlayan her farklı x ve y ada çifti için, x ve y arasında yelken açmak için kullanılacak tam olarak iki kano vardır. Bunlardan biri x adasına, diğeri ise y adasına demirlenmiştir.
3. (21 puan) $N \leq 1000$, M çift sayıdır. $0 \leq i \leq M - 1$ şartını sağlayan her i çift sayısı için, i ve $i + 1$ kanolarının her ikisi de $U[i]$ ve $V[i]$ adaları arasında yelken açmak için kullanılabilir. Kano i başlangıçta $U[i]$ adasında ve kano $i + 1$ başlangıçta $V[i]$ adasındadır. Biçimsel olarak, $U[i] = V[i + 1]$ ve $V[i] = U[i + 1]$.

4. (24 puan) $N \leq 1000$, M çift sayıdır. $0 \leq i \leq M - 1$ şartını sağlayan her i çift sayısı için, i ve $i + 1$ kanolarının her ikisi de $U[i]$ ve $V[i]$ adaları arasına yelken açmak için kullanılabilir. Her iki kano da başlangıçta $U[i]$ adasındadır. Biçimsel olarak, $U[i] = U[i + 1]$ ve $V[i] = V[i + 1]$.
5. (45 puan) Ek kısıt yoktur.

Geçerli bir yolculuğun bulunduğu her test senaryosu için çözümünüz:

- geçerli bir yolculuk return ederse tam puan alır,
- true, 2 000 000'dan fazla tamsayı elemanı olan bir dizi veya geçerli bir yolculuğu tanımlamayan bir dizi return ederse, puanın 35% 'ini alır,
- aksi takdirde 0 puan alır.

Geçerli bir yolculuğun bulunmadığı her test senaryosu için çözümünüz:

- false return ederse tam puan alır,
- aksi takdirde 0 puan alır.

Her alt görev için nihai puanın, alt görevdeki test senaryoları için minimum puan olduğuna dikkat edin.

Örnek Değerlendirici

Örnek değerlendirici girdiyi aşağıdaki formatta okur:

- satır 1: $N M$
- satır $2 + i$ ($0 \leq i \leq M - 1$): $U[i] V[i]$

Örnek değerlendirici cevabınızı aşağıdaki formatta yazar:

- Eğer `find_journey`, `bool` return ederse:
 - satır 1: 0
 - satır 2: 0 eğer `find_journey`, `false` return ederse, ya da 1 diğer durumda.
- `find_journey` bir `int[]` return ederse, bu dizinin elemanlarını sırasıyla $c[0], c[1], \dots, c[k - 1]$ ile belirtin. Örnek değerlendirici şunu yazar:
 - satır 1: 1
 - satır 2: k
 - satır 3: $c[0] c[1] \dots c[k - 1]$