

## Anahtarlar

Bilinen bir mimar olan Timothy yeni bir odadan kaçış oyunu tasarlıyor. Bu oyunda  $0$  ile  $n - 1$  arasında numaralandırılmış  $n$  adet oda bulunmakta. Başlangıçta her bir odada tam olarak bir tane anahtar mevcut. Her bir anahtarın  $0$  ile  $n - 1$  ( $0$  ve  $n - 1$  dahil) arasında tam sayı olarak verilmiş özel bir türü olduğu biliniyor.  $r[i]$  değeri  $i$  ( $0 \leq i \leq n - 1$ ) odasındaki anahtarın türünü belirtiyor. Birden fazla odada aynı türe sahip anahtar olabiliyor, yani,  $r[i]$  değerlerinin tamamen farklı olması şart değil.

Ayrıca bu oyunda  $0$  ile  $m - 1$  arasında numaralandırılmış  $m$  adet **çift yönlü** birleştirici bulunuyor.  $j$  ( $0 \leq j \leq m - 1$ ) birleştiricisi iki farklı odayı,  $u[j]$  ve  $v[j]$ , birbirine bağlamaktadır. Oda ikilileri birden fazla birleştirici ile birbirine bağlanmış olabilir.

Oyun odalardan anahtarları toplayıp, birleştiriciler vasıtasıyla odalar arasında geçiş yapabilen tek bir oyuncu tarafından oynanmaktadır. Oyuncu  $j$  numaralı birleştiriciyi kullanırsa,  $u[j]$  odasından  $v[j]$  odasına veya  $v[j]$  odasından  $u[j]$  odasına **geçiş yapabilecektir**. Fakat oyuncu sadece  $c[j]$  türündeki anahtara sahip olduğu takdirde  $j$  numaralı birleştiriciyi kullanabilir.

Oyunun herhangi bir anında, oyuncunun herhangi bir  $x$  odasında olduğunu varsaydığımızda, oyuncu şu iki hamleyi yapabilir:

- $x$  odasındaki anahtarı alabilir, bu anahtarın türü  $r[x]$  olacaktır (eğer bu anahtara zaten sahip değilse),
- $j$  numaralı birleştiriciden geçebilir, bu durumda  $u[j] = x$  veya  $v[j] = x$  olmalıdır. Tabi ki oyuncunun daha önce  $c[j]$  türündeki anahtara sahip olması gereklidir. Dikkate alınması gereken nokta sahip olunan anahtarın kullanımının **hiçbir zaman** son bulmamasıdır.

Oyuncu oyuna  $s$  odasında hiçbir anahtarı olmadan **başlamaktadır**. Eğer oyuncu  $s$  odasından oyuna başladığında, verilen tanımlı hamleleri yaparak  $t$  odasına varabiliyorsa,  $t$  odası  $s$  odasından **ulaşılabilir**dir.

Her bir  $i$  ( $0 \leq i \leq n - 1$ ) odası için  $i$  odasından ulaşılabilir tüm odalar  $p[i]$  olarak tanımlanmıştır. Timothy,  $0 \leq i \leq n - 1$  arasında, en küçük  $p[i]$  değerine sahip olan  $i$  değerlerini bulmak istiyor.

## Implementasyon Detayları

Aşağıda verilen fonksiyonu kodlamanız gerekmektedir:

```
int[] find_reachable(int[] r, int[] u, int[] v, int[] c)
```

- $r$ :  $n$  uzunluğunda dizi. Her bir  $i$  ( $0 \leq i \leq n - 1$ ) değeri için,  $i$  odasındaki anahtarın türü olan  $r[i]$  değeridir.

- $u, v$ :  $m$  uzunluğunda iki adet dizi. Her bir  $j$  ( $0 \leq j \leq m - 1$ ) değeri için,  $j$  numaralı birleştirici  $u[j]$  ve  $v[j]$  odalarını birbirine bağlar.
- $c$ :  $m$  uzunluğunda bir dizi. Her bir  $j$  ( $0 \leq j \leq m - 1$ ) değeri için,  $j$  numaralı birleştiriciyi kullanabilmek için gerekli olan anahtar türü  $c[j]$  değeridir.
- Bu fonksiyon  $n$  uzunluğunda  $a$  dizisini döndürmelidir. Her bir  $0 \leq i \leq n - 1$  değeri için, eğer tüm  $j$   $0 \leq j \leq n - 1$  değerleri için,  $p[i] \leq p[j]$  ise,  $a[i]$  değeri 1 olmalıdır. Aksi takdirde  $a[i]$  değeri 0 olacaktır.

## Örnekler

### Örnek 1

Aşağıdaki fonksiyonun çağırıldığını varsayalım:

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

Eğer oyuncu oyuna 0 numaralı odadan başlıyorsa, sırasıyla aşağıdaki hamleleri yapabilir:

Bulunduğu oda	Hamle
0	0 türündeki anahtarı al
0	Birleştirici 0 yoluyla 1 numaralı odaya geçiş yap
1	1 türündeki anahtarı al
1	Birleştirici 2 yoluyla 2 numaralı odaya geçiş yap
2	Birleştirici 2 yoluyla 1 numaralı odaya geçiş yap
1	Birleştirici 3 yoluyla 3 numaralı odaya geçiş yap

Dolayısıyla 3 numaralı oda 0 numaralı odadan ulaşılabilir. Benzer bir şekilde diğer tüm odalar da 0 numaralı odadan ulaşılabilir, bu bize  $p[0] = 4$  değerini vermektedir. Aşağıdaki tablo tüm odalar için ulaşılabilir odaları vermektedir.

Başlangıç odası $i$	Ulaşılabilir odalar	$p[i]$
0	[0, 1, 2, 3]	4
1	[1, 2]	2
2	[1, 2]	2
3	[1, 2, 3]	3

Tüm odalar içinde en küçük  $p[i]$  değeri 2 olacaktır, bu değeri  $i = 1$  veya  $i = 2$  numaralı odalardan sağlayabiliriz. Dolayısıyla çağrılan fonksiyon  $[0, 1, 1, 0]$  sonucunu döndürmelidir.

## Örnek 2

```
find_reachable([0, 1, 1, 2, 2, 1, 2],  
               [0, 0, 1, 1, 2, 3, 3, 4, 4, 5],  
               [1, 2, 2, 3, 3, 4, 5, 5, 6, 6],  
               [0, 0, 1, 0, 0, 1, 2, 0, 2, 1])
```

Aşağıdaki tablo tüm odalar için ulaşılabilir odaları vermektedir.

Başlangıç odası $i$	Ulaşılabilir odalar	$p[i]$
0	[0, 1, 2, 3, 4, 5, 6]	7
1	[1, 2]	2
2	[1, 2]	2
3	[3, 4, 5, 6]	4
4	[4, 6]	2
5	[3, 4, 5, 6]	4
6	[4, 6]	2

Tüm odalar içinde en küçük  $p[i]$  değeri 2 olacaktır, bu değeri  $i \in \{1, 2, 4, 6\}$  odalardan sağlayabiliriz. Dolayısıyla çağrılan fonksiyon  $[0, 1, 1, 0, 1, 0, 1]$  sonucunu döndürmelidir.

## Örnek 3

```
find_reachable([0, 0, 0], [0], [1], [0])
```

Aşağıdaki tablo tüm odalar için ulaşılabilir odaları vermektedir.

Başlangıç odası $i$	Ulaşılabilir odalar	$p[i]$
0	[0, 1]	2
1	[0, 1]	2
2	[2]	1

Tüm odalar içinde en küçük  $p[i]$  değeri 1 olacaktır, bu değeri  $i = 2$  odasından sağlayabiliriz. Dolayısıyla çağrılan fonksiyon  $[0, 0, 1]$  sonucunu döndürmelidir.

## Kısıtlar

- $2 \leq n \leq 300\,000$
- $1 \leq m \leq 300\,000$

- bütün  $0 \leq i \leq n - 1$  için  $0 \leq r[i] \leq n - 1$
- bütün  $0 \leq j \leq m - 1$  için  $0 \leq u[j], v[j] \leq n - 1$  ve  $u[j] \neq v[j]$
- bütün  $0 \leq j \leq m - 1$  için  $0 \leq c[j] \leq n - 1$

## Altgörevler

1. (9 puan) bütün  $0 \leq j \leq m - 1$  ve  $n, m \leq 200$  için  $c[j] = 0$
2. (11 puan)  $n, m \leq 200$
3. (17 puan)  $n, m \leq 2000$
4. (30 puan) bütün  $0 \leq j \leq m - 1$  için  $c[j] \leq 29$  ve bütün  $0 \leq i \leq n - 1$  için  $r[i] \leq 29$
5. (33 puan) Ek kısıt bulunmamaktadır.

## Örnek Grader

Örnek grader girdiyi aşağıdaki formatta okumaktadır:

- satır 1:  $n \ m$
- satır 2:  $r[0] \ r[1] \ \dots \ r[n - 1]$
- satır  $3 + j$  ( $0 \leq j \leq m - 1$ ):  $u[j] \ v[j] \ c[j]$

Örnek grader `find_reachable` fonksiyonunun döndürdüğü değeri aşağıdaki formatta yazmaktadır:

- satır 1:  $a[0] \ a[1] \ \dots \ a[n - 1]$