



Gezilecek Yerleri Ayırma

Bakü'de 0'dan $n - 1$ 'e numaralandırılmış n tane gezilecek yer bulunmaktadır. Ayrıca 0'dan $m - 1$ 'e numaralandırılmış m tane her iki yönde gidilebilen yol bulunmaktadır. Her yol iki farklı gezilecek yeri birbirine bağlamaktadır. Verilen yollar kullanılarak herhangi bir gezilecek yerden diğerine gitmek mümkündür.

Fatima gezilecek bütün yerleri üç günde bitirmeyi planlamaktadır. Şimdiden birinci gün a tane, ikinci gün b tane ve üçüncü gün de c tane yeri gezmeye karar vermiştir. Bu nedenle, n gezilecek yeri A , B ve C kümeleri olarak birbirleri ile kesişmeyen ve büyüklükleri a , b ve c olan üç kümeye ayıracaktır. Her gezilecek yer, kümelerden yalnız birisine dahil olacaktır, yani $a + b + c = n$ 'dir.

Fatima öyle A , B ve C kümeleri bulmak istemektedir ki üç kümeden **en az iki tanesi bağlı** olsun. Bir S gezilecek yerler kümesine bağlı denmesi için S içindeki gezilecek yerlerden herhangi birisinden S içindeki başka bir gezilecek yere S içinde olmayan bir yerden geçmeden ve verilen yolları kullanarak gidilebilmesi gerekmektedir. Gezilecek yerlerin **geçerli** bir şekilde A , B ve C kümelerine ayrılması için yukarıda anlatılan şartların sağlanması gerekmektedir.

Verilen a , b ve c değerleri için Fatima'nın gezilecek yerleri geçerli bir şekilde kümelere ayırmasına yardım edin, ya da geçerli bir ayırımın mümkün olmadığını bulun. Birden fazla geçerli ayırım mümkün ise herhangi birisini bulmanız yeterli olacaktır.

Kodlama detayları

Aşağıdaki prosedürü kodlamanız gerekmektedir:

```
int[] find_split(int n, int a, int b, int c, int[] p, int[] q)
```

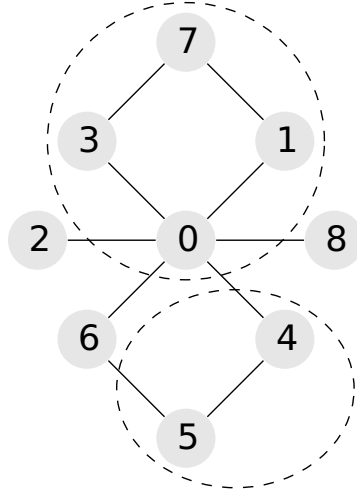
- n : gezilecek yerlerin sayısı.
- a , b ve c : A , B ve C kümelerinin istenilen boyutları.
- p ve q : m uzunluğunda yolların uç noktalarını içeren diziler. Her bir i ($0 \leq i \leq m - 1$) için, $p[i]$ ve $q[i]$, i yolu tarafından birbirine bağlanan iki gezilecek yerdir.
- Bu prosedür n uzunluğunda bir dizi döndürmelidir. Bu diziyeye s diyelim. Eğer geçerli bir kümelere ayırım mümkün değil ise, s dizisi n tane sıfır içermelidir. Eğer mümkün ise, $0 \leq i \leq n - 1$ için, $s[i]$ 'nin değeri ya 1, ya 2, ya da 3 olmalıdır ve bu değer i gezilecek yerinin sırasıyla A , B ya da C kümesine ait olduğunu gösterir.

Örnekler

Örnek 1

Aşağıdaki fonksiyon çağrımını göz önüne alalım:

```
find_split(9, 4, 2, 3, [0, 0, 0, 0, 0, 0, 1, 3, 4, 5],  
          [1, 2, 3, 4, 6, 8, 7, 7, 5, 6])
```

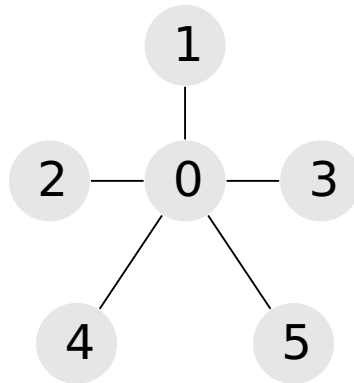


Olası bir doğru çözüm $[1, 1, 3, 1, 2, 2, 3, 1, 3]$ dizisidir. Bu çözüm şu küme ayrımını belirtmektedir: $A = \{0, 1, 3, 7\}$, $B = \{4, 5\}$ ve $C = \{2, 6, 8\}$. A ve B kümeleri bağlıdır.

Örnek 2

Aşağıdaki fonksiyon çağrımını göz önüne alalım:

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



Geçerli bir ayırım bulunmamaktadır. Bu nedenle, doğru cevap $[0, 0, 0, 0, 0, 0]$ olmalıdır.

Kısıtlar

- $3 \leq n \leq 100\,000$
- $2 \leq m \leq 200\,000$
- $1 \leq a, b, c \leq n$
- $a + b + c = n$
- Her bir gezilecek yer çifti arasında en fazla bir tane yol bulunmaktadır.
- Verilen yollar kullanılarak herhangi bir gezilecek yerden diğerine gitmek mümkündür.
- $0 \leq p[i], q[i] \leq n - 1$ ve $0 \leq i \leq m - 1$ için $p[i] \neq q[i]$

Altgörevler

1. (7 puan) Her bir gezilecek yer en fazla iki yolun uç noktasıdır.
2. (11 puan) $a = 1$
3. (22 puan) $m = n - 1$
4. (24 puan) $n \leq 2500, m \leq 5000$
5. (36 puan) Ek kısıt bulunmamaktadır.

Örnek puanlayıcı

Örnek puanlayıcı girdiyi aşağıdaki formatta okur:

- satır 1: $n \ m$
- satır 2: $a \ b \ c$
- satır $3 + i$ ($0 \leq i \leq m - 1$ için): $p[i] \ q[i]$

Örnek puanlayıcı `find_split` çağrısının döndüğü diziyi tek bir satırda ekrana basar.