



# En Uzun Seyahat

IOI 2023 organizatörlerinin başı büyük derttedir! Önümüzdeki gün için Ópusztaszer gezisini planlamayı unuttular. Ama belki de henüz çok geç değildir..

Ópusztaszer'de 0'dan  $N - 1$ 'a kadar indislenmiş  $N$  tane turistik bölge vardır. Bu bölgelerin bazı çiftleri *çift yönlü yollarla* birbirine bağlıdır. Her bir bölge çifti en fazla bir yolla birbirine bağlıdır. Organizatörler hangi bölgelerin birbirine bağlı olduğunu *bilmiyor*.

Ópusztaszer'deki yol ağı **yoğunluğu**, herhangi 3 farklı bölgenin arasında en az  $\delta$  yolun olduğu durumda **en az**  $\delta$  olarak ifade edilir. Başka bir deyişle,  $0 \leq u < v < w < N$  koşulunu sağlayan her üçlü bölge  $(u, v, w)$  için  $(u, v)$ ,  $(v, w)$  ve  $(u, w)$  bölge çiftleri arasında en az  $\delta$  çift birbirine bağlıdır.

Organizatörler, yol ağı yoğunluğunun en az  $D$  olduğu pozitif bir  $D$  tamsayısını *biliyor*.

$D$  değerinin 3 'den büyük olamayacağını unutmayın.

Organizatörler, belirli bölgeler arasındaki yol bağlantıları hakkında bilgi toplamak için Ópusztaszer'deki merkezi **arayabilir**. Her çağrıda,  $[A[0], \dots, A[P - 1]]$  ve  $[B[0], \dots, B[R - 1]]$  bölgelerinin boş olmayan iki dizisi belirtilmelidir. Turistik bölgeler ikili olarak farklı olmalıdır, yani

- Her  $i$  ve  $j$  için  $A[i] \neq A[j]$  öyle ki  $0 \leq i < j < P$ ;
- Her  $i$  ve  $j$  için  $B[i] \neq B[j]$  öyle ki  $0 \leq i < j < R$ ;
- $0 \leq i < P$  ve  $0 \leq j < R$  olacak şekilde her  $i$  ve  $j$  için  $A[i] \neq B[j]$ .

Her çağrı için merkez,  $A$  bölgesini  $B$  bölgesine bağlayan bir yol olup olmadığını bildirir. Yani,  $0 \leq i < P$  ve  $0 \leq j < R$  olacak şekilde  $A[i]$  ve  $B[j]$ 'yi bağlayacak  $i$  ve  $j$  varsa merkez `true` değerini döner. Aksi taktirde merkez `false` değerini döner.

$l$  uzunluğundaki bir **yolculuk**,  $t[0], t[1], \dots, t[l - 1]$  farklı bölgelerden oluşan bir seridir; burada 0 ve  $l - 2$  arasında (sınırlar dahil) her  $i$  için bölge  $t[i]$  ile bölge  $t[i + 1]$  bir yolla birbirine bağlıdır.  $l$  uzunluğundaki bir yolculuğa, eğer en az  $l + 1$  uzunluğunda bir yolculuk yoksa, **en uzun yolculuk** adı verilir.

Göreviniz, merkeze çağrı yaparak organizatörlerin Ópusztaszer'deki en uzun yolculuğu bulmalarına yardımcı olmaktır.

## Kodlama Detayları

Aşağıdaki prosedürü kodlamalısınız:

```
int[] longest_trip(int N, int D)
```

- $N$ : Ópusztaszer'daki turistik bölge sayısı.
- $D$ : Yol ağının garanti edilen minimum yoğunluğu.
- Bu prosedür en uzun yolculuğu temsil eden  $t = [t[0], t[1], \dots, t[l - 1]]$  dizisini dönmelidir.
- Bu prosedür her bir test senaryosunda **birden fazla** çağrılabilir.

Yukarıdaki prosedür aşağıdaki prosedürü çağırabilir:

```
bool are_connected(int[] A, int[] B)
```

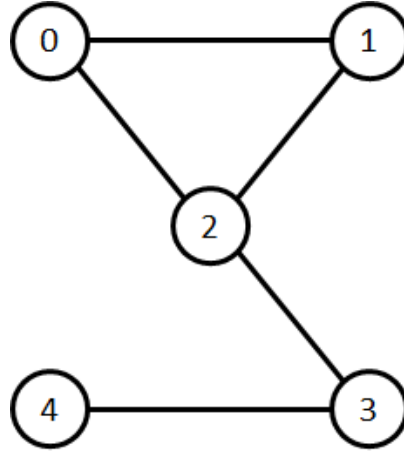
- $A$ : Turistik bölgeleri içeren dizi. Bu dizi boş değildir.
- $B$ : Turistik bölgeleri içeren dizi. Bu dizi boş değildir.
- $A$  ve  $B$  ayrık olmalıdır.
- Eğer  $A$ 'daki bir bölge ve  $B$ 'deki bir bölge bir yol ile birbirine bağlı ise bu prosedür `true` döner. Diğer türlü `false` döner.
- Bu prosedür, her `longest_trip` çağrısında en fazla 32 640 kez ve toplamda en fazla 150 000 kez çağrılabilir.
- Bu prosedüre tüm çağrıları boyunca iletilen  $A$  ve  $B$  dizilerinin toplam uzunluğu 1 500 000'u aşamaz.

Grader **adaptif(uyarlanabilir) değildir**. Her gönderim aynı test senaryoları seti üzerinden puanlandırılır. Yani,  $N$  ve  $D$  değerlerinin yanı sıra yollarla birbirine bağlanan bölge çiftleri, her test senaryosu içindeki her `longest_trip` çağrısı için sabitlenir.

## Örnekler

### Örnek 1

$N = 5$ ,  $D = 1$  olan ve yol bağlantılarının aşağıdaki şekilde gösterildiği gibi olduğu bir senaryoyu düşünün:



longest\_trip prosedürü aşağıdaki şekilde çağrılır:

```
longest_trip(5, 1)
```

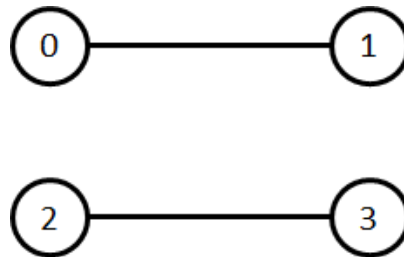
Bu prosedür aşağıdaki gibi are\_connected 'a çağrı yapabilir.

Çağrı	Yol ile bağlı çiftler	Return değeri
are_connected([0], [1, 2, 4, 3])	(0,1) ve (0,2)	true
are_connected([2], [0])	(2,0)	true
are_connected([2], [3])	(2,3)	true
are_connected([1, 0], [4, 3])	hiçbiri	false

Dördüncü çağrıdan sonra, (1,4), (0,4), (1,3) ve (0,3) çiftlerinden *hiçbirinin* bir yol ile bağlı olmadığı ortaya çıkar. Ağın yoğunluğu en az  $D = 1$  olduğundan, (0,3,4) üçlüsünden (3,4) çiftinin bir yolla bağlanması gerektiğini görüyoruz. Buna benzer şekilde, 0 ve 1 bölgeleri bağlanmalıdır.

Bu noktada  $t = [1,0,2,3,4]$ 'nin 5 uzunluğunda bir yolculuk olduğu ve 5'den daha uzun bir yolculuğun mevcut olmadığı sonucuna varılabilir. Bu nedenle, longest\_trip prosedürü  $[1,0,2,3,4]$  değerini dönebilir.

$N = 4$ ,  $D = 1$  olan ve bölgeler arasındaki yolların aşağıdaki şekilde gösterildiği gibi olduğu başka bir senaryoyu düşünün:



longest\_trip prosedürü aşağıdaki gibi çağrılır:

```
longest_trip(4, 1)
```

Bu senaryoda en uzun yolculuğun uzunluğu 2'dir. Bu nedenle, are\_connected prosedürüne yapılan birkaç çağrıdan sonra, longest\_trip prosedürü [0,1], [1,0], [2,3] veya [3,2] 'den birini dönebilir.

## Örnek 2

Alt görev 0,  $N = 256$  bölgeye sahip ek bir örnek test senaryosu içerir. Bu test senaryosu yarışma sisteminden indirebileceğiniz ek pakette yer almaktadır.

## Kısıtlar

- $3 \leq N \leq 256$
- longest\_trip 'e yapılan tüm  $N$  çağrının toplamı her test senaryosu içinde 1 024'ü aşmaz.
- $1 \leq D \leq 3$

## Altgörevler

1. (5 puan)  $D = 3$
2. (10 puan)  $D = 2$
3. (25 puan)  $D = 1$ .  $l^*$  en uzun yol uzunluğunu gösterebilir. longest\_trip prosedürünün  $l^*$  uzunluğunda bir yolculuk dönmesi gerekmez. Bunun yerine, en az  $\left\lceil \frac{l^*}{2} \right\rceil$  uzunluğunda bir yolculuk dönmelidir.
4. (60 puan)  $D = 1$

Alt görev 4'te puanınız, tek bir longest\_trip çağrısı üzerinden are\_connected prosedürüne yapılan çağrılarının sayısına göre belirlenir.  $q$ , bu alt görevin her test senaryosunda tüm longest\_trip çağrısı arasındaki maksimum çağrı sayısı olsun. Bu alt göreve ilişkin puanınız aşağıdaki tabloya göre hesaplanır:

Şart	Puan
$2\,750 < q \leq 32\,640$	20
$550 < q \leq 2\,750$	30
$400 < q \leq 550$	45
$q \leq 400$	60

Test senaryolarının herhangi birinde, are\_connected prosedürüne yapılan çağrılar Kodlama Detaylarında açıklanan kısıtlamalara uymuyorsa veya longest\_trip tarafından dönen dizi yanlışsa, o alt görev için çözümünüz 0 puan alacaktır.

## Örnek Değerlendirici (Sample Grader)

$C$  senaryoların sayısını, yani `longest_trip` 'e yapılan çağrıların sayısını gösterebilir. Örnek değerlendirici girdiyi aşağıdaki formatta okur:

- satır 1:  $C$

$C$  senaryonun açıklamaları takip eder.

Örnek değerlendirici her senaryonun açıklamasını aşağıdaki formatta okur:

- satır 1:  $N$   $D$
- satır  $1 + i$  ( $1 \leq i < N$ ):  $U_i[0] U_i[1] \dots U_i[i - 1]$

Burada, her  $U_i$  ( $1 \leq i < N$ ), hangi bölge çiftlerinin bir yolla birbirine bağlandığını tanımlayan,  $i$  büyüklüğünde bir dizidir.  $1 \leq i < N$  ve  $0 \leq j < i$  olacak şekilde her  $i$  ve  $j$  için:

- $j$  ve  $i$  bölgeleri bir yolla birbirine bağlıysa, o zaman  $U_i[j]$  değeri 1 olmalıdır;
- $j$  ve  $i$  bölgelerini birbirine bağlayan bir yol yoksa, o zaman  $U_i[j]$  değeri 0 olmalıdır.

Her senaryoda, `longest_trip` çağrısını yapmadan önce örnek değerlendirici, yol ağı yoğunluğunun en az  $D$  olup olmadığını kontrol eder. Bu koşul sağlanmadığı takdirde `Insufficient Density` (Yetersiz Yoğunluk) mesajını basar ve sonlanır.

Örnek değerlendirici bir protokol ihlali tespit ederse, örnek değerlendiricinin çıktısı `Protocol Violation: <MSG>` olur; burada `<MSG>` aşağıdaki hata mesajlarından biridir:

- `invalid array`: `are_connected` çağrısında,  $A$  ve  $B$  dizilerinin en az biri
  - boştur, ya da
  - 0 ile  $N - 1$  (sınırlar dahil) arasında tam sayı olmayan bir eleman içerir, ya da
  - aynı elemanı en az iki kere içerir.
- `non-disjoint arrays`: `are_connected` çağrısında,  $A$  ve  $B$  dizileri ayrık değildir.
- `too many calls`: `are_connected` 'a yapılan çağrıların sayısı, mevcut `longest_trip` çağrısına göre 32 640'ı veya toplamda 150 000'i aşmaktadır.
- `too many elements`: tüm çağrılar göz önüne alındığında `are_connected` 'a gönderilen bölgelerin toplam sayısı 150 000'i aşmaktadır.

Aksi takdirde, bir senaryoda `longest_trip` tarafından dönen dizinin elemanları, negatif olmayan bir  $l$  için  $t[0], t[1], \dots, t[l - 1]$  olsun. Örnek değerlendirici bu senaryo için aşağıdaki formatta üç satır yazar:

- satır 1:  $l$
- satır 2:  $t[0] t[1] \dots t[l - 1]$
- satır 3: bu senaryoda `are_connected` 'a yapılan çağrıların sayısı

Son olarak örnek değerlendirici şunları yazar:

- satır  $1 + 3 \cdot C$ : longest\_trip 'a yapılan tüm çağrılar üzerinden are\_connected 'a yapılan maksimum çağrı sayısı