

Sihirli Şehir

Taşkent belediye başkanı, ünlü Sihirli Şehir eğlence parkını yeniden tasarlamak istiyor. Size K pozitif tamsayısı verilmiştir ve göreviniz parkı şu şekilde tasarlamaktır:

- Parktaki atraksiyonların sayısını belirlemek üzere N şeklinde bir tamsayı seçin. Atraksiyonlar, 0 ile $N - 1$ arasındaki tamsayılarla numaralandırılmıştır.
- Her biri **farklı** iki atraksiyon arasında çift yönlü geçit (walkway) ekleyin. Aynı atraksiyon çifti arasında birden fazla geçit bulunabilir. **Geçitleri kullanarak her atraksiyon çifti arasında yolculuk yapılabilmesi zorunlu değildir.**
- $0 \leq i < N$ koşulunu sağlayan her i için, i atraksiyonuna $T[i]$ değerini tür olarak atayın. 0 ile $2K - 1$ arasında numaralandırılmış $2K$ tür vardır. Her tür en az bir atraksiyona atanmalıdır. Farklı atraksiyonlar aynı türe sahip olabilir.

Pazar araştırması şu sonuçları ortaya koydu:

- Her ziyaretçi, aynı türden iki atraksiyon ardı ardına gelmeyecek şekilde üç atraksiyonu ziyaret etmeyi tercih eder.
- Ziyaretçiler, birçok geçitin uç noktası olan atraksiyonlardan hoşlanmazlar.

Tüm potansiyel ziyaretçileri memnun etmek amacıyla, belediye başkanı tasarımınıza iki koşul getirmiştir.

Tiplerden oluşan sıralı bir üçlüye (t_1, t_2, t_3) , $0 \leq t_1, t_2, t_3 < 2K$ **ilginç** diyoruz; eğer $t_1 \neq t_2$ ve $t_2 \neq t_3$ ise. t_1 'in t_3 'e eşit olabileceğine dikkat edin. Dolayısıyla, $2K \cdot (2K - 1)^2$ ilginç üçlü vardır.

Koşul 1: Her ilginç üçlü (t_1, t_2, t_3) için, aşağıdaki koşulları sağlayan üç atraksiyon a_1, a_2, a_3 ($0 \leq a_1, a_2, a_3 < N$ ile) bulunmalıdır:

- a_1, a_2, a_3 türleri, t_1, t_2, t_3 ile eşleşmelidir. Yani, $T[a_1] = t_1$, $T[a_2] = t_2$ ve $T[a_3] = t_3$ olmalıdır.
- a_1 ile a_2 arasında bir geçit bulunmalıdır.
- a_2 ile a_3 arasında bir geçit bulunmalıdır.

a_1 ile a_3 arasında bir geçit olup olmadığı önemli değildir. Ayrıca, $t_1 = t_3$ olduğunda, a_1 ve a_3 atraksiyonlarının aynı olabileceğini de unutmayın.

Koşul 2: Her atraksiyon, **en fazla** K geçitin uç noktası olabilir.

Bu görev, **kısmi puanlama (partial scoring)** uygulanan 50 **output-only** alt görevinden oluşur. Her alt görev, K değerlerinden birine karşılık gelir ve siz, o K değeri için yukarıdaki tüm koşulları

karşılayan bir park tasarlamalısınız. Puanınız, çözümünüzdeki atraksiyon sayısına bağlıdır: daha az atraksiyon, daha yüksek veya eşit bir puan sağlar.

Kodlama Detayları

Çözümünüzü göndermek için iki yöntem vardır ve her alt görev için bunlardan birini kullanabilirsiniz:

- **Procedure call**
- **Output file**

Çözümünüzü **procedure call** yoluyla göndermek için aşağıdaki prosedürü kodlamalısınız:

```
std::pair<std::vector<int>, std::vector<std::pair<int, int>>>>  
construct(int K)
```

- K : atraksiyon türlerinin sayısının yarısı ve ayrıca her bir atraksiyonun uç noktası olabileceği maksimum geçit sayısı.
- Bu prosedür, her alt görev için tam olarak bir kez çağrılır.

Prosedür, bir eğlence parkını tanımlayan bir çift (T, E) döndürmelidir. M , parkınızdaki geçit sayısı olsun.

- T : atraksiyon türlerini tanımlayan, uzunluğu N olan bir dizi.
- E : geçitleri tanımlayan, uzunluğu M olan bir dizi. Her bir $0 \leq j < M$ için, $E[j] = (U[j], V[j])$, farklı atraksiyonlar $U[j]$ ve $V[j]$ arasındaki çift yönlü bir geçiti temsil eder.

Çözümünüzü bir **output file** aracılığıyla göndermek için, aşağıdaki formatta bir metin dosyası oluşturup gönderin:

```
N M  
T[0] T[1] ... T[N-1]  
U[0] V[0]  
U[1] V[1]  
...  
U[M-1] V[M-1]
```

Çözümünüzün **geçerli** sayılabilmesi için aşağıdaki kısıtlamaları karşılaması gerektiğini unutmayın:

- $N \leq 2000$
- $0 \leq T[i] < 2K$ her bir $0 \leq i < N$ için, ve her tür en az bir atraksiyona atanmalıdır.
- $0 \leq U[j], V[j] < N$ ve $U[j] \neq V[j]$ her bir $0 \leq j < M$ için.
- Koşul 1 ve 2 karşılanmalıdır.

Kısıtlamalar

- $1 \leq K \leq 50$

Puanlama

50 alt görev vardır; her biri 1 ile 50 aralığındaki K tamsayıları için birer alt görevdir. i ($1 \leq i \leq 50$) alt görevi için, K değeri i 'dir.

Her alt görevin, aşağıdaki tabloya göre bir puanı S ve bir hedef atraksiyon sayısı P vardır.

Alt görevler	S	P
1	1	2
2	8	12
3	9	24
4	9	40
5	9	50
6 - 10	4	$12 \cdot K$
11 - 12	3	$12 \cdot K$
13 - 50	1	$12 \cdot K$

Her bir alt görev için, çözümünüz geçerli bir eğlence parkını tanımlamıyorsa, çözümünüzün puanı 0 olacaktır (CMS'de `Output isn't correct` olarak bildirilir).

Aksi takdirde, puanınız N ve S ve P parametrelerine göre aşağıdaki şekilde hesaplanır:

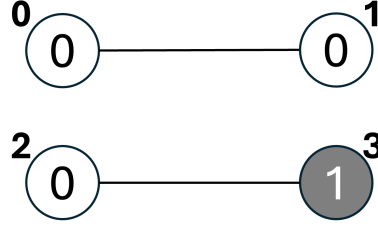
Koşul	Puan
$N \leq P$	S
$P < N \leq 2P$	$(0.4 + 0.3 \cdot \frac{2P-N}{P}) \cdot S$
$2P < N \leq 2000$	$(0.1 + 0.3 \cdot \frac{2P}{N}) \cdot S$

Örnek

Aşağıdaki çağırışı ele alalım:

```
construct (1)
```

Bu örnekte, $K = 1$ olduğundan, $2K = 2$ türde atraksiyon vardır. Aşağıdaki şekil, $N = 4$ atraksiyon ve $M = 2$ geçit içeren geçerli bir çözümü göstermektedir. 0,1,2 numaralı atraksiyonları 0 türündedir ve 3 numaralı atraksiyon 1 türündedir.



İki ilginç üçlü vardır:

- $(0, 1, 0)$ tür üçlüsü için $(a_1, a_2, a_3) = (2, 3, 2)$ seçebiliriz.
- $(1, 0, 1)$ tür üçlüsü için $(a_1, a_2, a_3) = (3, 2, 3)$ seçebiliriz.

Bu, Koşul 1'in karşılandığı anlamına gelir.

Prosedür, $([0, 0, 0, 1], [(0, 1), (2, 3)])$ çiftini döndürebilir. Bu örnek için sunulan çözümün, $K = 1$ için en uygun çözüm olmayabileceğine dikkat edin.

Örnek Değerlendirici

Girdi biçimi:

K

Çıktı biçimi:

```
N M
T[0] T[1] ... T[N-1]
U[0] V[0]
U[1] V[1]
...
U[M-1] V[M-1]
```

Örnek değerlendiricinin çıktısının, çıktı dosyası için gerekli biçimle uyumlu olduğuna dikkat edin.