

Sihir Gösterisi

Alice ve Bob çok ünlü sihirbazlardır. Catherine ise onların yaptıkları bu işe çok değer veren varlıklı bir kadındır. Catherine bu sihirbazlardan kendisi için bir sihir yapmalarını istemektedir. Sihrin içeriği aşağıdaki gibidir:

- Adım 1: Bob bir kabine girer ve sadece Catherine ile iletişim kurabilir. Daha sonra Alice Catherine'e bir sayı söyler. n 2 ile 5 000 arasındadır.
- Adım 2: Catherine Alice'e bir sayı söyler X 1 ile 10^{18} arasındadır.
- Adım 3: Alice n düğümlü bir ağaç oluşturur ve onu Catherine'e verir.
- Adım 4: Catherine en fazla $\lfloor \frac{n-2}{2} \rfloor$ kenarı siler ve kalan kenarları Bob'a teslim eder.
- Adım 5: Bob çizgeyi dikkatlice inceler ve Catherine'nin Alice'e verdiği sayıyı (x)'i söyler.

Alice ve Bob bu sihri yapabilmek için gerekli olan zekaya sahip olduklarını düşünmedikleri için sizden yardım istiyorlar. Hem Alice için hem de Bob için strateji geliştirerek Catherine'nin istediği sihrin gerçekleşmesine yardımcı olun.

Implementation Details

İki adet dosya göndermeniz gerekmektedir:

İlk dosya Alice'in stratejisini kodlayan Alice.cpp olmalıdır. Kodunuz Alice.h kütüphanesini içermelidir. #include kullanarak kütüphaneyi ekleyiniz. Bu kodda kodlanması gereken fonksiyon aşağıdaki gibidir:

```
std::vector<std::pair<int, int>> Alice();
```

- Her test durumu için fonksiyon sadece bir kere çağırılır (başlangıçta).
- Fonksiyon içinde pair içeren bir vektör dönmelidir. Bu vektör Alice'in 3.adımda oluşturduğu ağacın kenarlarına sahiptir.
 - Düğümlerin 1'den başlayarak numaralandırıldığına dikkat ediniz.
 - Döndürdüğünüz çizgenin bir ağaç olması gerektiğine dikkat ediniz, kısaca $n - 1$ kenar olmalı ve bütün düğümler bağlı bir yapı içerisinde bulunmalıdır (connected).

Alice() fonksiyonu aşağıdaki fonksiyonu **tam olarak** bir kere çağırmalıdır:

```
long long setN(int n);
```

- Bu fonksiyon sayesinde n belirlenir, Alice'in Catherine'e 1.adımda verdiği sayı.
- Bu fonksiyon X sayısını döner, Catherine'in Alice'e 2.adımda verdiği sayı.

İkinci dosya ise Bob'un stratejisini kodlayan Bob.cpp olacak. Bob.h kütüphanesini #include kullanarak eklemeyi unutmayınız. Dosyada kodlanması gereken fonksiyon aşağıdaki gibidir:

```
long long Bob(std::vector<std::pair<int, int>> V);
```

- Bu fonksiyon her test durumu için Alice() fonksiyonu çağırıldıktan sonra tam olarak bir kere çağırılacaktır.
- V Catherine'nin Bob'a 4. Adımda verdiği çizgenin kenarlarını(edge) içeren bir listedir.
- Kenarlar size **sıralı bir şekilde** verilecektir. Sıralı şu anlama gelir:
 - Her kenar için uçlarındaki düğümlerden küçük olan solda olacak;
 - Kenarlar önce soldaki düğüme bakarak sıralanacak, eşitlik durumunda ise sağdaki düğüme bakılacak. Küçük olan daha yukarıda temsil edilecek.
- Fonksiyon X sayısını döner.

Sample Interaction

Call	Return Value
Alice()	
setN(4)	3
	{{1,2},{2,3},{2,4}}
Bob({{1,2},{2,4}})	3

Aşağıdaki senaryoyu temsil eder:

- Adım 1: Başta, Alice 4 sayısını Catherine'e verir.
- Step 2: Catherine 3 sayısını Alice'e verir.
- Step 3: Alice 4 düğümlü {{1,2},{2,3},{2,4}} ağacını oluşturur ve Catherine'e verir.
- Step 4: Catherine 2 ve 3'ü bağlayan kenarı siler ve kalan kenarları: {{1,2},{2,4}} Bob'a verir.
- Step 5: Bob 3 sayısını söyler. Cevap doğru olduğu için sihir başarıyla gerçekleşmiştir.

Constraints

- $1 \leq X \leq 10^{18}$.

Subtasks

1. (5 points): $X \leq 5\,000$.
2. (30 points): $X \leq 25\,000\,000$.

3. (65 points): Hiçbir ek kısıt yoktur.

Sample Grader

The sample grader reads the input in the following format:

- Line 1: T ($T \in \{1, 2\}$)

if $T = 1$, then the sample grader reads as follows:

- Line 2: X ($1 \leq X \leq 10^{18}$)

The sample grader prints your answer of function `Alice()` in the following format:

- Line 1: n
- Line $2 + i$ ($0 \leq i \leq n - 2$): $u[i] \ v[i]$, where there exists an edge connecting $u[i], v[i]$.

if $T = 2$, then The sample grader reads as follows:

- Line 2: $n \ m$ ($2 \leq n \leq 5000, n - 1 - \lfloor \frac{n-2}{2} \rfloor \leq m \leq n - 1$), where n is number of vertices, and m is number of remaining edges.
- Line $3 + i$ ($0 \leq i \leq m - 1$): $u[i] \ v[i]$, which means there exists an edge connecting $u[i], v[i]$.

The sample grader prints your answer of function `Bob()` in the following format:

- Line 1: X