

## Pertunjukan Sulap

Alice dan Bob merupakan pesulap terkenal. Catherine, seorang wanita kaya yang tertarik dengan kelihaian mereka, menyatakan bahwa ia akan memberikan kekayaan yang berlimpah kepada mereka apabila mereka dapat melakukan trik sulap berikut. Isinya adalah sebagai berikut:

- Langkah 1: Bob masuk ke dalam sebuah ruangan dan terisolasi dari luar. Bob hanya dapat berkomunikasi dengan Catherine. Kemudian, Alice memberitahu Catherine sebuah bilangan bulat  $n$  di antara 2 dan 5 000.
- Langkah 2: Catherine memberitahukan sebuah bilangan bulat  $X$  kepada Alice, yang berada di antara 1 dan  $10^{18}$ .
- Langkah 3: Alice membuat sebuah *tree* yang terdiri dari tepat  $n$  buah *node*, dan memberikannya kepada Catherine.
- Langkah 4: Catherine menghapus paling banyak  $\lfloor \frac{n-2}{2} \rfloor$  buah *edge* dari *tree* tersebut, dan memberikan semua *edge* yang tersisa kepada Bob.
- Langkah 5: Bob mencermati graf tersebut dan menebak bilangan bulat yang Catherine telah berikan kepada Alice.

Namun, Alice dan Bob merasa mereka tidak cukup ahli agar dapat berhasil melakukan trik sulap ini, jadi mereka meminta bantuan Anda. Buatlah sebuah program yang mengimplementasikan strategi Alice dan strategi Bob sehingga mereka dapat menyelesaikan tantangan Catherine.

## Detail Implementasi

Anda harus mengumpulkan **dua berkas**:

Berkas pertama adalah `Alice.cpp`, yang mengimplementasikan strategi Alice. Berkas ini harus memuat `Alice.h` menggunakan direktif praproses `#include`. Fungsi yang harus diimplementasikan pada berkas ini adalah:

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

- Untuk setiap kasus uji, fungsi ini dipanggil tepat satu kali di awal.
- Fungsi ini harus mengembalikan sebuah *vector* berisi *pair*, yang merepresentasikan semua *edge* pada *tree* yang Alice buat pada Langkah 3 dari trik sulap tersebut.
  - Perhatikan bahwa *node* dari *tree* harus dinomori dari 1.
  - Anda harus memastikan bahwa *tree* yang dikembalikan sesuai, yakni harus terdapat tepat  $n - 1$  buah *edge* dan semua *node* harus saling terhubung.

Fungsi `Alice()` harus memanggil fungsi berikut **tepat satu kali**:

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

- Dengan fungsi ini, Alice memilih parameter  $n$  yang ia berikan kepada Catherine pada Langkah 1 dari trik sulap tersebut.
- Fungsi ini kemudian mengembalikan nilai  $X$ , yang Catherine berikan kepada Alice pada Langkah 2 dari trik sulap tersebut.

Berkas kedua adalah `Bob.cpp`, yang mengimplementasikan strategi Bob. Berkas ini harus memuat `Bob.h` menggunakan direktif praproses `#include`. Fungsi yang harus diimplementasikan pada berkas ini adalah:

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

- Untuk setiap kasus uji, fungsi ini dipanggil tepat satu kali setelah pemanggilan fungsi `Alice()`.
- Parameter  $V$  merupakan daftar *edge* dari graf yang Catherine berikan kepada Bob pada Langkah 4 dari trik sulap tersebut.
- Semua *edge* diberikan **secara terurut**, yang berarti:
  - Untuk kedua ujung dari setiap *edge*, *node* dengan nomor yang lebih kecil diberikan terlebih dahulu;
  - Semua *edge* diberikan secara terurut menaik berdasarkan ujung pertama sebagai kata kunci pertama dan ujung kedua sebagai kata kunci kedua.
- Fungsi ini harus mengembalikan sebuah bilangan bulat, yang merepresentasikan bilangan  $X$ .

## Contoh Interaksi

Pemanggilan	Nilai kembalian
<code>Alice()</code>	
<code>setN(4)</code>	3
	$\{\{1, 2\}, \{2, 3\}, \{2, 4\}\}$
<code>Bob(\{\{1, 2\}, \{2, 4\}\})</code>	3

Ini merepresentasikan skenario berikut:

- Langkah 1: Pertama-tama, Alice memberikan 4 kepada Catherine.
- Langkah 2: Catherine memberikan 3 kepada Alice.
- Langkah 3: Alice membuat sebuah *tree* dengan 4 *node* dan *edge* berikut  $\{\{1, 2\}, \{2, 3\}, \{2, 4\}\}$ , dan memberikannya kepada Catherine.

- Langkah 4: Catherine menghapus *edge* yang menghubungkan *node* 2 dan 3, dan memberikan semua *edge* yang tersisa  $\{\{1, 2\}, \{2, 4\}\}$  kepada Bob.
- Langkah 5: Bob menebak 3. Karena jawabannya benar, maka mereka berhasil melakukan pertunjukan sulap tersebut.

## Batasan

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

## Subsoal

1. (5 poin):  $X \leq 5\,000$ .
2. (30 poin):  $X \leq 25\,000\,000$ .
3. (65 poin): Tidak ada batasan tambahan.

## Contoh Grader

Contoh *grader* membaca masukan dengan format berikut:

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

jika  $T = 1$ , maka contoh *grader* membaca masukan berikut:

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

Contoh *grader* mencetak jawaban Anda dari fungsi `Alice()` dengan format berikut:

- Baris 1:  $n$
- Baris  $2 + i$  ( $0 \leq i \leq n - 2$ ):  $u[i] \ v[i]$ , dimana terdapat suatu *edge* yang menghubungkan  $u[i], v[i]$ .

jika  $T = 2$ , maka contoh *grader* membaca masukan berikut:

- Baris 2:  $n \ m$  ( $2 \leq n \leq 5000, n - 1 - \lfloor \frac{n-2}{2} \rfloor \leq m \leq n - 1$ ), dengan  $n$  adalah banyaknya *node*, and  $m$  adalah banyaknya *edge* yang tersisa.
- Baris  $3 + i$  ( $0 \leq i \leq m - 1$ ):  $u[i] \ v[i]$ , yang berarti terdapat suatu *edge* yang menghubungkan  $u[i], v[i]$ .

Contoh *grader* mencetak jawaban Anda dari fungsi `Bob()` dengan format berikut:

- Baris 1:  $X$