

Show Mágico

Alicia y Robertico son dos magos famosos. Caterina, una mujer acaudalada se interesó bastante por sus increíbles hazañas y prometió darles una gran recompensa si podían hacer el siguiente truco de magia. El truco consiste en los siguientes pasos:

- Paso 1: Robertico entra a una habitación y se desconecta completamente del exterior. Robertico solo puede comunicarse con Caterina. Entonces, Alicia le dice a Caterina un número n entre 2 y 5000.
- Paso 2: Caterina le dice un número X a Alicia, el cual debe estar entre 1 y 10^{18} .
- Paso 3: Alicia hace un árbol con exactamente n vértices, y se lo da a Caterina.
- Paso 4: Caterina elimina a lo más $\lfloor \frac{n-2}{2} \rfloor$ aristas del árbol, y le da las aristas restantes a Robertico.
- Paso 4: Robertico observa cuidadosamente el grafo, y le dice a Caterina el número que le dijo a Alicia.

Sin embargo, Alicia y Robertico creen que no son lo suficientemente inteligentes para llevar con éxito este truco de magia, por lo que están buscando tu ayuda. Por favor escribe un programa que implemente la estrategia de Alicia y la de Robertico de tal forma que puedan vencer el reto de Caterina.

Detalles de Implementación

Debes enviar **dos archivos**:

El primer archivo, cuyo nombre es `Alice.cpp`, implementa la estrategia de Alicia. En él, tienes que incluir la librería `Alice.h` usando el comando `#include`. La función que debes implementar en este archivo es:

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

- Para cada caso de prueba, esta función es llamada exactamente una vez, al principio.
- La función debe regresar un vector de pares, que representan las aristas del árbol que Alicia construye en el Paso 3 del truco.
 - Observa que los vértices del árbol deben estar numerados a partir del 1.
 - Debes asegurarte que el árbol que regreses sea correcto, lo que significa que debe tener exactamente $n - 1$ aristas y que todos los vértices deben estar conectados.

La función `Alice()` debe llamar la siguiente función **exactamente una vez**:

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

- Usando esta función, Alicia elige el parámetro n que le da a Caterina en el Paso 1 del truco.
- La función regresa el valor X , que Caterina le da a Alicia en el Paso 2 del truco.

El segundo archivo, cuyo nombre es `Bob.cpp`, implementa la estrategia de Robertico. En él, tienes que incluir la librería `Bob.h` usando el comando `#include`. La función que debes implementar en este archivo es:

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

- Para cada caso de prueba, esta función es llamada exactamente una vez, después de la llamada a la función `Alice()`.
- El primer parámetro V es la lista de aristas del grafo que Caterina le da a Robertico en el Paso 4 del truco.
- Las aristas son dadas **en orden ascendente**, lo que significa que:
 - Para cualquier arista, el vértice numerado menor aparece primero;
 - Todas las aristas están ordenadas en orden ascendente primero por su primer vértice, y luego por el segundo.
- La función debe regresar un solo entero, que representa el número X .

Ejemplo de interacción

Llamada	Valor de regreso
<code>Alice()</code>	
<code>setN(4)</code>	3
	$\{\{1, 2\}, \{2, 3\}, \{2, 4\}\}$
<code>Bob(\{\{1, 2\}, \{2, 4\}\})</code>	3

El ejemplo representa el siguiente escenario:

- Paso 1: Primero, Alicia le da el número 4 a Caterina.
- Paso 2: Caterina le da el número 3 a Alicia.
- Paso 3: Alicia hace un árbol con 4 vértices, aristas $\{\{1, 2\}, \{2, 3\}, \{2, 4\}\}$, y se lo dice a Caterina.
- Paso 4: Caterina le quita la arista conectando los vértices 2 y 3, y le da las aristas resultantes $\{\{1, 2\}, \{2, 4\}\}$ a Robertico.

- Paso 5: Robertico dice el número 3. Como la respuesta es correcta, ellos ejecutan el truco exitosamente.

Límites

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

Subtareas

1. (5 puntos): $X \leq 5\,000$.
2. (30 puntos): $X \leq 25\,000\,000$.
3. (65 puntos): Sin restricciones adicionales.

Evaluador de Ejemplo

El evaluador de ejemplo lee la entrada en el siguiente formato:

- Línea 1: T ($T \in \{1, 2\}$)

Si $T = 1$, entonces el evaluador de ejemplo lee como sigue:

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

El evaluador de ejemplo imprime tu respuesta de la función `Alice()` en el siguiente formato:

- Línea 1: n
- Línea $2 + i$ ($0 \leq i \leq n - 2$): $u[i] \ v[i]$, donde existe una arista conectando los nodos $u[i], v[i]$.

Si $T = 2$, entonces el evaluador de ejemplo lee como sigue:

- Línea 2: $n \ m$ ($2 \leq n \leq 5000, n - 1 - \lfloor \frac{n-2}{2} \rfloor \leq m \leq n - 1$), donde n es el número de vértices, y m es el número de aristas restantes.
- Línea $3 + i$ ($0 \leq i \leq m - 1$): $u[i] \ v[i]$, lo que significa que existe una arista que conecta $u[i], v[i]$.

El evaluador de ejemplo imprime tu respuesta de la función `Bob()` en el siguiente formato:

- Línea 1: X