

# Magic Show

Алиса и Боб - знаменитые фокусники. Богатая женщина Кэтрин, заинтересовавшаяся их удивительными делами, заявила, что подарит им огромное богатство, если они смогут выполнить следующий фокус. Содержание фокуса следующее:

- Шаг 1: Боб входит в комнату и полностью отрезается от внешнего мира. Боб может общаться только с Кэтрин. Затем Алиса говорит Кэтрин число  $n$  от 2 до 5 000.
- Шаг 2: Кэтрин сообщает Алисе число  $X$ , которое находится в диапазоне от 1 до  $10^{18}$ .
- Шаг 3: Алиса строит дерево с ровно  $n$  вершинами и отдает его Кэтрин.
- Шаг 4: Екатерина удаляет не более  $\lfloor \frac{n-2}{2} \rfloor$  ребер из дерева, а оставшиеся ребра отдает Бобу.
- Шаг 5: Боб внимательно наблюдает за графом и называет число, которое Кэтрин сказала Алисе.

Однако Алиса и Боб не считают себя достаточно умными, чтобы успешно выполнить этот фокус, поэтому они обращаются к вам за помощью. Пожалуйста, напишите программу, которая реализует стратегию Алисы и Боба, чтобы они смогли решить задачу Кэтрин.

## Implementation Details

Вам необходимо отослать **два файла**:

Первый файл - `Alice.cpp`, который реализует стратегию Алисы. Он должен включать `Alice.h` с помощью директивы препроцессорирования `#include`. Функция, которую необходимо реализовать в этом файле, следующая:

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

- Для каждого тестового случая эта функция вызывается ровно один раз в начале.
- Функция должна возвращать вектор пар, которые представляют собой ребра в дереве Алисы, построенном на Шаге 3.
  - Обратите внимание, что вершины дерева должны быть пронумерованы, начиная с 1.
  - Необходимо убедиться, что возвращаемое граф является деревом, то есть в нем должно быть ровно  $n - 1$  ребер и все вершины должны быть связными.

Функция `Alice()` должна вызывать следующую функцию **ровно один раз**:

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

- Используя эту функцию, Алиса выбирает параметр  $n$ , который она передала Кэтрин в Шаг 1.
- Затем функция возвращает значение  $X$ , которое Кэтрин передала Алисе на Шаг 2.

Второй файл - Bob.cpp, который реализует стратегию Боба. Он должен включать Bob.h с помощью директивы препроцессорирования #include. Функция, которую необходимо реализовать в этом файле, следующая:

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

- Для каждого тестового случая эта функция вызывается ровно один раз после вызова функции Alice().
- Параметр  $V$  - это список ребер графа, который Кэтрин передала Бобу на Шаг 4.
- Ребра даны **в отсортированном порядке**, что означает:
  - Для двух конечных вершин каждого ребра меньшая по номеру вершина стоит первой;
  - Все ребра отсортированы в порядке возрастания сначала по первой вершине ребра, при их равенстве по второй вершине.
- Функция должна возвращать одно целое число, которое представляет собой число  $X$ .

## Sample Interaction

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

Это представляет следующее:

- Шаг 1: Сначала Алиса отдает 4 Кэтрин.
- Шаг 2: Кэтрин отдает 3 Алисе.
- Шаг 3: Алиса строит дерево с 4 вершинами и ребрами  $\{\{1, 2\}, \{2, 3\}, \{2, 4\}\}$ , и передает его Кэтрин.
- Шаг 4: Кэтрин вырезает ребро, соединяющее вершины 2 и 3, и отдает оставшиеся ребра  $\{\{1, 2\}, \{2, 4\}\}$  Бобу.
- Шаг 5: Боб называет число 3. Поскольку его ответ верен, они могут успешно провести магический трюк.

## Constraints

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

## Subtasks

1. (5 баллов):  $X \leq 5\,000$ .
2. (30 баллов):  $X \leq 25\,000\,000$ .
3. (65 баллов): Нет дополнительных ограничений.

## Sample Grader

Образец грайдера считывает входные данные в следующем формате:

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

если  $T = 1$ , тогда образец грайдер считывает данные в следующем формате:

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

Образец грайдера выводит ваш ответ на функцию `Alice()` в следующем формате:

- Строка 1:  $n$
- Строка  $2 + i$  ( $0 \leq i \leq n - 2$ ):  $u[i] \ v[i]$ , где существует ребро, соединяющее  $u[i], v[i]$ .

если  $T = 2$ , тогда образец грайдер считывает данные в следующем формате:

- Строка 2:  $n \ m$  ( $2 \leq n \leq 5000, n - 1 - \lfloor \frac{n-2}{2} \rfloor \leq m \leq n - 1$ ), где  $n$  количество вершин, и  $m$  количество ребер.
- Строка  $3 + i$  ( $0 \leq i \leq m - 1$ ):  $u[i] \ v[i]$ , что означает, что существует ребро, соединяющее  $u[i], v[i]$ .

Образец грайдера выводит ваш ответ на функцию `Bob()` в следующем формате:

- Строка 1:  $X$