

Фокусники

Алиса и Боб — известные маги. Катерина, богатая женщина, которая заинтересовалась их трюками, объявила, что она может гарантировать солидную спонсорскую поддержку, если они смогут выполнить следующий фокус:

- Шаг 1: Боб заходит в комнату, из которой не может никаким образом общаться с кем-то, кроме Катерины. Затем Алиса сообщает Катерине целое число n между 2 и 5 000.
- Шаг 2: Катерина сообщает Алисе целое число X между 1 и 10^{18} .
- Шаг 3: Алиса строит дерево с ровно n вершинами и передаёт его Катерине.
- Шаг 4: Катерина удаляет из дерева не более $\lfloor \frac{n-2}{2} \rfloor$ рёбер, после чего сообщает информацию про оставшиеся ребра Бобу.
- Шаг 5: Боб изучает полученный граф и называет число, которое Катерина сообщила Алисе.

Алиса и Боб не уверены, что они смогут выполнить фокус, и просят вас помочь им. Напишите программу, которая реализует стратегию Алисы и Боба для того, чтобы успешно выполнить задание Катерины.

Детали реализации

Вы должны отправить **два файла**:

Первый файл — `Alice.cpp`, который реализует стратегию Алисы. В нём должен быть подключен заголовочный файл `Alice.h` с помощью директивы препроцессора `#include`. Функция, которая должна быть реализована в этом файле:

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

- Для каждого теста функция вызывается ровно один раз в начале.
- Функция должна вернуть вектор пар, который задаёт рёбра в дереве, которое Алиса построила на третьем шаге фокуса.
 - Вершины дерева должны быть пронумерованы последовательными целыми числами, начиная с 1.
 - Вы должны гарантировать, что передаваемое делево корректно, то есть что в графе ровно $n - 1$ ребро и граф является связным.

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

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

- Эта функция используется для того, чтобы Алиса сообщила параметр n , который она выбирает и передаёт Катерине на первом шаге фокуса.
- Функция возвращает число X , которое Катерина даёт Алисе на втором шаге фокуса.

Второй файл — `Bob.cpp`, который реализует стратегию Боба. В нём должен быть подключён заголовочный файл `Bob.h` с помощью директивы препроцессора `#include`. Функция, которая должна быть реализована в этом файле:

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

- Для каждого теста эта функция вызывается один раз сразу после вызова функции `Alice()`.
- Параметр V — список рёбер графа, которые Катерина передаёт Бобу на четвёртом шаге фокуса.
- Рёбра даны **в отсортированном порядке**, что означает:
 - из двух вершин, соединённых ребром, вершина с меньшим номером идёт первой;
 - рёбра отсортированы по возрастанию первой вершины, а при равенстве — по возрастанию второй.
- Функция возвращает целое число, которое должно быть равно числу X из второго шага фокуса.

Пример взаимодействия

Вызов	Возвращаемое значение
<code>Alice()</code>	
<code>setN(4)</code>	3
	$\{\{1, 2\}, \{2, 3\}, \{2, 4\}\}$
<code>Bob(\{\{1, 2\}, \{2, 4\}\})</code>	3

Пример соответствует следующему сценарию:

- Шаг 1: В начале фокуса Алиса сообщает Катерине число 4.
- Шаг 2: Катерина сообщает Алисе число 3.
- Шаг 3: Алиса строит дерево с 4 вершинами и рёбрами $\{\{1, 2\}, \{2, 3\}, \{2, 4\}\}$, после чего передаёт его Катерине.
- Шаг 4: Катерина убирает ребро, соединяющее вершины 2 и 3, и даёт оставшиеся рёбра $\{\{1, 2\}, \{2, 4\}\}$ Бобу.
- Шаг 5: Боб сообщает число 3 Катерине. Так как ответ правильный, фокус удался.

Ограничения

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

Подзадачи

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

Грейдер для участника

Предоставляемый участникам грейдер принимает входные данные в следующем формате:

- Строка 1: T ($T \in \{1, 2\}$)

если $T = 1$, то далее следует:

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

Грейдер выводит ответ, который вернула ваша реализация функции `Alice()`, в следующем формате:

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

Если $T = 2$, то далее следуют:

- 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, а m — количество оставшихся рёбер.
- Строка $3 + i$ ($0 \leq i \leq m - 1$): $u[i] \ v[i]$, обозначающая очередное оставшееся ребро, соединяющее вершины $u[i], v[i]$.

Грейдер выводит ответ, который вернула ваша реализация функции `Bob()`, в следующем формате:

- Строка 1: X