

C. Fun Tour

Time limit	2 s
Memory limit	512 MB

Описание

В самом большом тематическом парке Джакарты есть N достопримечательностей, пронумерованных от 0 до $N - 1$. Эти достопримечательности соединены $N - 1$ двусторонними дорогами таким образом, что существует ровно один способ добраться от одной достопримечательности до другой по дорогам. Дороги пронумерованы от 0 до $N - 2$. Дорога i соединяет достопримечательности $A[i]$ и $B[i]$, чтобы пройти по дороге требуется один час. Чтобы избежать столпотворений, каждая достопримечательность является концом не более чем трех дорог.

Вы хотели бы прогуляться по парку, посетив каждую достопримечательность ровно один раз. Но идти по большому числу дорог от одной достопримечательности до другой ужасно скучно. Чтобы прогулка была интересной, вы бы хотели посетить достопримечательности в таком порядке, чтобы переход к очередной достопримечательности занимал не больше, чем переход к предыдущей. Другими словами, вы хотите выбрать такую перестановку $P[0], P[1], \dots, P[N - 1]$, содержащую все числа от 0 до $N - 1$ ровно по одному разу, что время, необходимое для перехода от достопримечательности $P[i]$ к достопримечательности $P[i + 1]$, было не больше, чем время, необходимое для перехода от достопримечательности $P[i - 1]$ к достопримечательности $P[i]$ для всех $0 < i < N - 1$.

У вас нет полной карты достопримечательностей. Поэтому вам придется задать несколько вопросов информационному центру парка, чтобы спланировать интересную прогулку. Вы можете задать не более Q вопросов, каждый из которых имеет два параметра X и Y , где $0 \leq X, Y < N$. Каждый вопрос должен быть одного из следующих типов:

- Сколько часов требуется, чтобы дойти от достопримечательности X до достопримечательности Y ? В частности, если $X = Y$, от ответ на этот вопрос 0.
- Сколько существует достопримечательностей Z , таких, что если вы будете идти от достопримечательности X до достопримечательности Z , то вы пройдете через достопримечательность Y ? Сама достопримечательность Y также считается, поэтому, например, если $X = Y$, то ответ N .

Задание

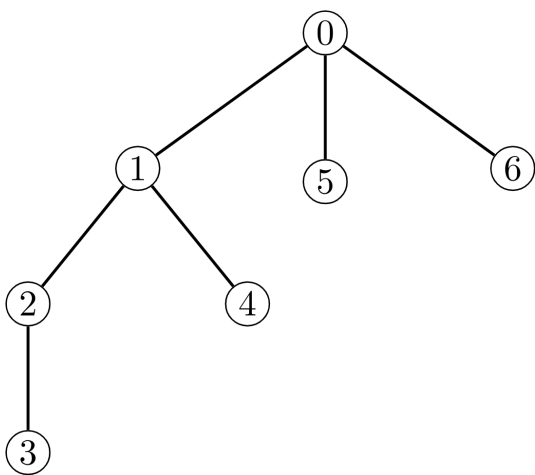
Вам необходимо реализовать функцию `createFunTour`:

- `createFunTour(N, Q)` - Эта функция будет вызвана проверяющим модулем ровно один раз.
 - N : Целое число - количество достопримечательностей.
 - Q : Целое число - максимальное количество вопросов.
 - В этой функции вы можете задавать вопросы, вызывая две функции проверяющего модуля:
 - `hoursRequired(X, Y)`
 - X : Целое число - первая достопримечательность.
 - Y : Целое число - вторая достопримечательность.
 - Эта функция возвращает целое число, равное количеству часов, которое требуется, чтобы дойти от достопримечательности X до достопримечательности Y .
 - Если хотя бы одно из чисел X или Y не лежит в диапазоне от 0 до $N - 1$, вы получите вердикт WA.

- `attractionsBehind(X, Y)`
 - X : Целое число - первая достопримечательность.
 - Y : Целое число - вторая достопримечательность.
 - Функция возвращает целое число, равное количеству достопримечательностей Z , таких, что если вы будете идти от достопримечательности X до достопримечательности Z , то вы пройдете через достопримечательность Y .
 - Если хотя бы одно из чисел X или Y не лежит в диапазоне от 0 до $N - 1$, вы получите вердикт WA.
- Функция должна вернуть массив из N целых чисел, задающих перестановку достопримечательностей, в которой их необходимо посетить во время интересной прогулки.

Пример

В следующем примере, $N = 7$, $Q = 400\,000$, $A = [0, 0, 0, 1, 1, 2]$ и $B = [1, 5, 6, 2, 4, 3]$. Пример проиллюстрирован на следующем рисунке:



Проверяющий модуль вызовет функцию `createFunTour(7, 400000)`.

- Если программа участника сделает вызов `hoursRequired(3, 5)`, функция вернет 4.
- Если программа участника сделает вызов `hoursRequired(5, 4)`, функция вернет 3.
- Если программа участника сделает вызов `attractionsBehind(5, 1)`, функция вернет 4. Чтобы попасть от пятой достопримечательности к первой, второй, третьей и четвертой, необходимо пройти через первую достопримечательность.
- Если программа участника сделает вызов `attractionsBehind(1, 5)`, функция вернет 1.
- В качестве результата работы можно вернуть $[3, 6, 4, 5, 2, 0, 1]$ в этом случае переходы к очередной достопримечательности будут занимать, соответственно, $[4, 3, 3, 3, 2, 1]$.

Ограничения

- $2 \leq N \leq 100\,000$.
- $Q = 400\,000$.
- Существует ровно один способ добраться от одной достопримечательности до другой по дорогам.
- Каждая достопримечательность является концом не более чем трех дорог.

Подзадача 1 (10 баллов)

- $N \leq 17$.

Подзадача 2 (16 баллов)

- $N \leq 500$.

Подзадача 3 (21 балл)

- Существует дорога, соединяющая достопримечательность i с достопримечательностью $\lfloor \frac{i-1}{2} \rfloor$ для всех $1 \leq i < N$.

Подзадача 4 (19 баллов)

- Есть хотя бы одна достопримечательность T , такая что для всех $0 \leq i < N$, `hoursRequired(T, i) < 30` и есть интервал $[L[i], R[i]]$ ($0 \leq L[i] \leq i \leq R[i] < N$), удовлетворяющий следующим ограничениям:
 - вы должны посетить достопримечательность i на пути от достопримечательности T до достопримечательности j , если и только если $L[i] \leq j \leq R[i]$.
 - Если $L[i] < i$, то существует ровно одна достопримечательность X , такая что:
 - $L[i] \leq X < i$.
 - Есть дорога, соединяющая достопримечательность i с достопримечательностью X .
 - Если $i < R[i]$, то существует ровно одна достопримечательность Y , такая что:
 - $i < Y \leq R[i]$.
 - Есть дорога, соединяющая достопримечательность i с достопримечательностью Y .

Подзадача 5 (34 балла)

- Нет дополнительных ограничений.

Пример проверяющего модуля

Пример проверяющего модуля читает входные данные из стандартного потока ввода в следующем формате:

```
N Q
A[0] B[0]
A[1] B[1]
.
.
.
A[N-2] B[N-2]
```

Пример проверяющего модуля выводит массив, который возвращает функция `createFunTour`, если она корректно вернула массив из N целых чисел, они соответствуют перестановке достопримечательностей в интересной прогулке и суммарное количество вызовов `hoursRequired` и `attractionsBehind` не превышает Q . Иначе выводится сообщение об ошибке.