

C. 즐거운 행로

Time limit	2 s
Memory limit	512 MB

설명

자카르타의 제일 큰 놀이동산에는 N 가지 놀이기구가 있는데, 0부터 $N - 1$ 까지 번호가 매겨져 있다. 이 탈 것들은 $N - 1$ 개의 양 방향 도로로 연결되어 있어서, 어떤 두 놀이기구를 골라도 이 둘을 잇는 유일한 경로가 존재한다. 이 도로는 0부터 $N - 2$ 까지 번호가 매겨져 있다. i 번 도로는 놀이기구 $A[i]$ 와 놀이기구 $B[i]$ 를 연결하고 걸어서 이동하는데 한 시간이 걸린다. 혼잡을 막기 위해서, 모든 놀이기구는 최대 3개의 도로와 연결되어 있다.

모든 놀이기구를 정확하게 한 번 방문하는 행로(tour)를 만들려고 한다. 한 놀이기구에서 다른 놀이기구로 이동할 때 여러 도로를 지나가는 것은 지루하다. 즐거운 행로를 만들려면, 모든 놀이기구에 대해서 순서 관계를 정해서, 다음 놀이기구를 방문하는데 필요한 시간이 직전 놀이기구를 방문하는데 필요한 시간보다 길지 않게 하고 싶다. 다른 말로 하면, 0부터 $N - 1$ 까지 모든 정수가 정확하게 한 번씩 나오는 순열 $P[0], P[1], \dots, P[N - 1]$ 을 찾으려 하는데, 모든 $0 < i < N - 1$ 에 대해서 놀이기구 $P[i]$ 에서 놀이기구 $P[i + 1]$ 로 이동하는데 걸리는 시간이 놀이기구 $P[i - 1]$ 에서 놀이기구 $P[i]$ 로 이동하는데 걸리는 시간보다 길지 않다.

당신은 놀이기구의 전체 지도를 가지고 있지 않다. 따라서, 즐거운 행로를 만들려면 안내센터에 질문을 여러번 해야 한다. 최대 Q 번 질문할 수 있고, 각각의 질문은 두 파라미터 X 와 Y 로 이루어지는데, $0 \leq X, Y < N$ 이다. 각 질문은 다음 둘 중 하나이다.

- 놀이기구 X 에서 놀이기구 Y 로 이동하는데 몇 시간 걸리는가? 특히 $X = Y$ 일 때, 답은 0이다.
- 놀이기구 X 와 놀이기구 Y 가 주어졌을 때, 다음 조건을 만족하는 놀이기구 Z 가 몇 개 있는가? 놀이기구 X 에서 놀이기구 Z 로 이동하려면 놀이기구 Y 를 반드시 방문해야 한다. 놀이기구 Y 도 포함된다. 특히 $X = Y$ 일 때, 답은 N 이다.

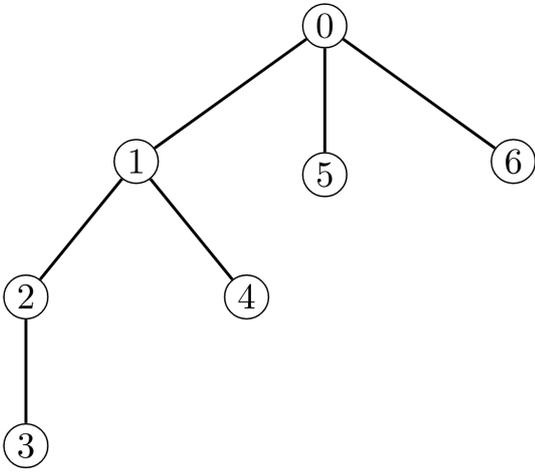
할 일

다음 `createFunTour` 함수를 구현해야 한다.

- `createFunTour(N, Q)` - 이 함수는 그레이더에 의해서 정확하게 한 번 호출된다.
 - N : 놀이기구의 수를 나타내는 정수.
 - Q : 질문의 최대 횟수를 나타내는 정수.
 - 이 함수는 다음 두 그레이더 함수를 호출할 수 있다.
 - `hoursRequired(X, Y)`
 - X : 첫번째 놀이기구를 표현하는 정수.
 - Y : 두번째 놀이기구를 표현하는 정수.
 - 이 함수는 놀이기구 X 에서 놀이기구 Y 로 이동하는데 필요한 시간을 리턴한다.
 - 만약 X, Y 중 어느 하나, 또는 둘 모두가 0 이상 $N - 1$ 이하인 정수가 아니라면, **WA**를 받는다.
 - `attractionsBehind(X, Y)`
 - X : 첫번째 놀이기구를 표현하는 정수.
 - Y : 두번째 놀이기구를 표현하는 정수.
 - 이 함수는 다음 조건을 만족하는 놀이기구 Z 의 개수를 리턴하는데, 놀이기구 X 에서 놀이기구 Z 로 이동하려면 놀이기구 Y 를 반드시 방문해야 한다.
 - 만약 X, Y 중 어느 하나, 또는 둘 모두가 0 이상 $N - 1$ 이하인 정수가 아니라면, **WA**를 받는다.
 - 이 함수는 즐거운 행로를 표현하는 N 개의 정수의 순열을 저장한 배열을 리턴해야 한다.

예제

다음 예제에서, $N = 7$, $Q = 400\,000$, $A = [0, 0, 0, 1, 1, 2]$, $B = [1, 5, 6, 2, 4, 3]$ 이다. 이 예제는 다음 그림으로 설명할 수 있다.



그레이더는 `createFunTour(7, 400000)` 를 호출한다.

- 만약 여러분이 `hoursRequired(3, 5)` 로 질의하면, 리턴값은 4이다.
- 만약 여러분이 `hoursRequired(5, 4)` 로 질의하면, 리턴값은 3이다.
- 만약 여러분이 `attractionsBehind(5, 1)` 로 질의하면, 리턴값은 4이다. 놀이기구 5에서 놀이기구 1, 2, 3, 4로 가려면 놀이기구 1을 반드시 지나가야 한다.
- 만약 여러분이 `attractionsBehind(1, 5)` 로 질의하면, 리턴값은 1이다.
- 여러분의 가능한 리턴값 중 하나는 $[3, 6, 4, 5, 2, 0, 1]$ 인데, 다음 놀이기구를 방문하는데 필요한 시간은 차례대로 $[4, 3, 3, 3, 2, 1]$ 이기 때문이다.

제약 조건

- $2 \leq N \leq 100\,000$.
- $Q = 400\,000$.
- 어떤 두 놀이기구도 도로를 통해서 이동할 수 있다.
- 각 놀이기구는 최대 3개의 도로와 연결되어 있다. (즉, 최대 3개 도로의 끝점이다.)

Subtask 1 (10 points)

- $N \leq 17$.

Subtask 2 (16 points)

- $N \leq 500$.

Subtask 3 (21 points)

- 모든 $1 \leq i < N$ 에 대해서 놀이기구 i 와 놀이기구 $\lfloor \frac{i-1}{2} \rfloor$ 를 연결하는 도로가 있다.

Subtask 4 (19 points)

- 모든 $0 \leq i < N$ 에 대해서 `hoursRequired(T, i) < 30`를 만족하면서 다음 조건을 만족하는 구간 $[L[i], R[i]]$ ($0 \leq L[i] \leq i \leq R[i] < N$)이 존재하는 놀이도구 T 가 최소한 하나 존재한다.
 - 놀이기구 T 에서 놀이기구 j 로 이동하려면 놀이기구 i 를 방문해야 한다는 것과, $L[i] \leq j \leq R[i]$ 라는 것은 필요충분조건이다.
 - 만약 $L[i] < i$ 이라면, 다음 조건을 만족하는 놀이기구 X 가 정확히 하나 존재한다.
 - $L[i] \leq X < i$.
 - 놀이기구 i 와 놀이기구 X 를 연결하는 도로가 있다.
 - 만약 $i < R[i]$ 이라면, 다음 조건을 만족하는 놀이기구 Y 가 정확히 하나 존재한다.

- $i < Y \leq R[i]$.
- 놀이기구 i 와 놀이기구 Y 를 연결하는 도로가 있다.

Subtask 5 (34 points)

- 추가적인 제약 조건이 없다.

샘플 그레이더

샘플 그레이더는 입력을 다음 양식으로 읽는다.

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

샘플 그레이더는 만약 `createFunTour` 함수가 즐거운 행로를 표현하는 N 개의 정수의 순열을 저장한 배열을 정확히 구하고, `hoursRequired`, `attractionsBehind` 함수의 호출 횟수 합이 Q 이하이면 이 배열을 출력한다. 그렇지 않으면, 오답 메시지를 출력한다.