

## 악어의 지하 도시(Crocodile's Underground City)

고고학자 철수는 악어의 신비한 지하 도시를 탐험하다가 위험을 느끼고 도망치게 되었다. 지하 도시는  $N$  개의 방으로 구성되어 있다. 방들을 연결하는  $M$  개의 복도가 있는데, 각 복도는 서로 다른 두 방을 연결하며, 같은 쌍의 방을 연결하는 복도는 최대  $1$  개이다. 복도를 통과하는데 걸리는 시간은 복도마다 다를 수 있다.  $N$  개의 방들 중  $K$  개는 바로 탈출이 가능한 "탈출방"이다. 철수는 최초에 방  $0$  에 있다. 철수는 최대한 빨리 탈출방으로 가려고 한다.

악어 문지기는 철수가 탈출하는 것을 막으려고 한다. 문지기는 복도 중 임의의 하나를 막을 수 있는 방법이 있다. 어떤 시점이든 단 하나만 막을 수 있다. 즉, 문지기가 새로운 복도를 막으면 이전에 막았던 복도는 열린다.

철수의 상황을 좀더 자세히 설명하면 다음과 같다: 철수가 어떤 방을 떠나려고 할 때 문지기는 그 방과 연결된 복도들 중 하나를 막을 수 있다. 철수는 막히지 않은 복도 중 하나를 골라서 다른 방으로 이동한다. 철수가 복도에 일단 들어가고 나면 철수가 이동을 완료할 때까지는 문지기가 이 복도를 막을 수 없다. 철수가 다른 방에 들어가고 나면 문지기는 (방금 지나온 복도도 당연히 포함하여) 또 다른 복도를 막을 수 있고, 이런 식으로 반복된다.

철수는 탈출 계획을 미리 정해 놓기를 원한다. 정확히 말하자면, 각 방 마다, 그 방에 도착하면 어떤 식으로 행동해야 하는지의 계획이 미리 정해져 있어야 한다.  $A$  가 어떤 방이라고 하자. 만약  $A$  가 탈출방이라면 바로 탈출하면 되므로 별다른 계획이 필요하지 않다.  $A$  가 탈출방이 아니라면  $A$  에 대해서는 다음 중 한가지 형태의 계획이 있어야 한다.

- 만약  $A$  에 있다면 방  $B$  로 이동하는 것이 우선이다. 만약 그 쪽 복도가 막혀 있으면 방  $C$  로 이동하라.
- 이 계획 하에서는  $A$  에 절대 도달할 수 없으므로, 아무 계획이 없음.

어떤 경우에는 (예를 들어 어떤 계획하에서 철수가 싸이클을 도는 경우 등) 문지기가 탈출이 영원히 불가능하도록 만들 수도 있다는 점에 주의하라. 어떤 탈출 계획이 문지기가 어떤 작전을 쓰든지 철수가 유한한 시간 안에 탈출하는 것이 가능하다는 것을 보장하면 그 계획은 "좋은" 계획이라고 부른다. 어떤 좋은 계획 하에서, 철수가 그 시간이 지난 후에는 반드시 탈출한다고 보장할 수 있는 최소의 시간을  $T$  라고 하자. 그 경우, 그 좋은 계획의 "탈출 시간"은  $T$  라고 말한다.

## 해야 할 일

다음의 파라미터를 받는 함수 **travel\_plan(N, M, R, L, K, P)**를 작성하라.

- **N** - 방의 수. 방은 **0** 부터 **N-1** 까지 번호가 붙어 있다.
- **M** - 복도의 수. 복도는 **0** 부터 **M-1** 까지 번호가 붙어 있다.
- **R** - 복도들을 표현하는 정수의 2차원 배열. 각  $i(0 \leq i < M)$ 에 대해서 복도  $i$ 는 방  $R[i][0]$ 와 방  $R[i][1]$ 를 연결한다. 동일한 쌍의 방들을 연결하는 복도는 최대 하나이다.
- **L** - 복도를 통과하는데 걸리는 시간을 저장한 일차원 배열. 각  $i(0 \leq i < M)$ 에 대해서  $L[i](1 \leq L[i] \leq 1,000,000,000)$ 의 값은 철수가 복도  $i$ 를 통과하는 데 걸리는 시간이다.
- **K** - 탈출방의 수. ( $1 \leq K < N$ )
- **P** - 탈출방의 번호들을 가지는 크기 **K**인 정수의 일차원 배열. 각  $i(0 \leq i < K)$ 에 대해서  $P[i]$ 는  $i$ 번째 탈출방의 번호이다. 방 **0**는 탈출방에 포함되지 않는다. (**P**에 있는 값들은 모두 다르다.)

당신의 함수는 좋은 계획이 존재하는 탈출 시간의 최소값 **T**를 리턴해야 한다.

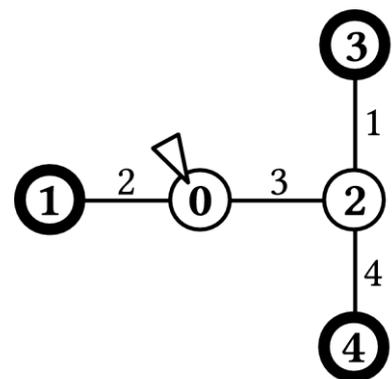
탈출방이 아닌 방은 최소한 **2**개의 복도가 연결되어 있다고 가정해도 좋다. 또한, 모든 경우에  $T \leq 1,000,000,000$  이하인 좋은 계획이 존재한다고 가정해도 좋다.

## 예제

### 예제 1

그림 1에서  $N=5, M=4, K=3$ 이고 **R, L, P**는 다음과 같다.

<b>R</b> =	0 1	2	1
	0 2	3	3
	3 2	1	4
	2 4	4	



방들은 원으로 표시되었고 복도는 줄로 표시되어 있다. 그림 1.

탈출방들은 굵은 원으로 표시되어 있다. 초기에 철수는 (삼각형으로 표시된) 방 **0**에 있다. 최적의 탈출 계획 중 하나는 다음과 같다.

- 방 **0**에 있다면, 방 **1**로 이동하는 것이 우선이다. 만약 그 방향이 막혀 있으면 방 **2**로

이동하라.

- 방 2 에 있다면, 방 3 으로 이동하는 것이 우선이다. 만약 그 방향이 막혀 있으면 방 4 로 이동하라.

최악의 경우에 시간 7 을 쓰고 탈출방에 도달할 수 있다. 따라서, **travel\_plan** 은 7 을 리턴해야 한다.

.

## 예제 2

그림 2 에서  $N=5$ ,  $M=7$ ,  $K=2$  이고  $R$ ,  $L$ ,  $P$  는 다음과 같다.

0 2	4	
0 3	3	
3 2	2	
R= 2 1	L= 10	P= 1
		3
0 1	100	
0 4	7	
3 4	9	

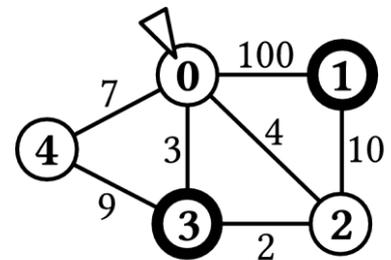


그림 2.

가능한 최적의 계획들 중 하나는 다음과 같다.

- 방 0 에 있으면 방 3 으로 이동하는 것이 우선이다. 만약 그 방향이 막혀 있으면 방 2 로 이동하라
- 방 2 에 있으면 방 3 으로 이동하는 것이 우선이다. 만약 그 방향이 막혀 있으면 방 1 로 이동하라
- 방 4 에는 가는 경우가 없으므로 아무 계획이 없다.

이 계획을 이용하면 철수는 시간 14 를 쓴 이후에는 반드시 탈출할 수 있다. 따라서 **travel\_plan** 은 14 를 리턴해야 한다.

## 태스크

### 서브태스크 1 (46 점)

- $3 \leq N \leq 1,000$ .
- 지하도시는 트리 모양이다. 즉,  $M = N-1$  이고 모든 방의 쌍  $i$  와  $j$  에 대해서 연결할 수 있는 경로가 있다.
- 모든 탈출방은 다른 하나의 방과만 직접 연결이 되어 있다.
- 모든 다른 방은 각각 3 개 이상의 방들과 직접 연결되어 있다.

### 서브태스크 2 (43 점)

- $3 \leq N \leq 1,000$ .
- $2 \leq M \leq 100,000$ .

### 서브태스크 3 (11 점)

- $3 \leq N \leq 100,000$ .
- $2 \leq M \leq 1,000,000$ .

## 구현시 유의사항

### 제약조건

- CPU 시간 제한: 2 초
- 메모리 제한: 256 MB

**유의사항:** 스택 메모리 크기에 대한 제한은 명시되지 않는다. 스택 메모리는 전체 메모리 사용량에 포함된다.

### 인터페이스 (API)

- 프로그램 작업폴더: crocodile/
- 참가자가 작성할 파일: crocodile.c 또는 crocodile.cpp 또는 crocodile.pas
- 참가자 인터페이스: crocodile.h 또는 crocodile.pas

- 채점프로그램(grader) 인터페이스: crocodile.h 또는 crocodilelib.pas
- 견본 채점프로그램(sample grader): grader.c 또는 grader.cpp 또는 grader.pas 와 crocodilelib.pas
- 견본 채점프로그램 입력(sample grader input): grader.in.1, grader.in.2, ...

**유의사항:** 견본 채점프로그램은 다음과 같은 양식으로 입력을 읽는다.:

- 1 번째 줄:  $N$ ,  $M$  과  $K$ .
- 2 번째 줄부터  $M+1$  번째 줄까지: 각  $i(0 \leq i < M)$ 에 대해  $i+2$  번째 줄에  $R[i][0]$ ,  $R[i][1]$ ,  $L[i]$ 가 공백으로 분리되어 주어진다.
- $M+2$  번째 줄:  $K$  개의 정수  $P[0]$ ,  $P[1]$ , ...,  $P[K-1]$ 가 공백을 사이에 두고 주어진다.
- $M+3$  번째 줄: 예상되는 정답.
- 견본 채점프로그램 입력에 대하여 예상되는 출력: grader.expect.1, grader.expect.2, ...

이 문제에 대해서 예상되는 출력 파일들은 정확히 다음의 내용을 가져야 한다:

**"Correct."** (마침표도 포함해야 함.)