

열대 식물원(Tropical Garden)

식물학자 철수는 여러 반의 학생들과 함께 태국 최대의 식물원을 방문하기로 하였다. 이 넓은 식물원은 (0 부터 $N-1$ 까지의 번호가 붙은) N 개의 연못과 M 개의 산책로로 구성되어 있다. 각 산책로는 서로 다른 두 연못을 연결하며, 양방향으로 모두 이동하는 것이 가능하다. 어느 연못이든지 최소 하나의 산책로가 연결되어 있다. 이 산책로들에는 철수가 좋아하는 아름다운 식물들이 많이 있다. 같은 반의 학생들은 함께 이동하며, 각 반이 산책을 시작하는 연못은 서로 다를 수 있다.

철수는 아름다운 열대 식물들을 아주 좋아한다. 따라서, 철수와 각 반의 학생들은 어느 연못에서든 가장 아름다운 산책로를 선택하여 이용한다. 단, 그 산책로를 바로 직전에 이용한 경우에는 두번째로 아름다운 산책로를 이용한다. 하지만, 현재 연못에 연결된 산책로가 단 하나뿐인 경우는 두 번째로 아름다운 산책로가 존재하지 않으므로 방금 사용한 산책로를 다시 이용하는 것을 허용한다. 철수는 식물학자이므로 두 산책로의 아름다운 정도가 같은 경우는 없다.

학생들은 식물에는 큰 관심이 없다. 학생들의 관심은 어떤 연못 P 옆에 위치한 고급 식당에서 점심 식사를 하는 것이다. 철수는 각 반의 학생들이 정확히 K 개의 산책로를 지난 다음 배가 고파질 것이라는 것을 알고 있다. 각 반 마다 K 의 값은 다를 수 있다. 이 상황 하에서 철수는 각 반에 대해 정확히 K 개의 산책로를 이용한 후 연못 P 에 도착하는 방법의 수가 얼마나 많은지 알고 싶다. 각 반에 대한 상황을 정리하면 다음과 같다

- 각 반은 아무 연못에서나 출발할 수 있다.
- 산책로를 선택하는 규칙은 위의 설명과 같다.
- 각 반은 정확히 K 개의 산책로를 이용한 다음에는 연못 P 에 도착해야 한다.

각 반이 P 로 가는 도중에 연못 P 를 지나는 것도 가능하다. 그러나, 마지막에는 반드시 P 에 도달하여야 한다.

해야 할 일

당신은 연못과 산책로에 대한 정보를 받아서 Q 개의 반에 대한 해답을 찾아야 한다. 즉, Q 개의 K 값에 대한 해답을 생성해야 하는 것이다.

다음의 입력을 받는 `count_routes(N, M, P, R, Q, G)` 함수를 작성하라.

- N - 연못의 수. 연못은 0 부터 $N-1$ 까지 번호가 붙어 있다.
- M - 산책로의 수. 산책로는 0 부터 $M-1$ 까지 번호가 붙어 있다. 산책로의 번호는 아름다운 정도가 줄어드는 순서이다. 즉, 모든 $i(0 \leq i < M-1)$ 에 대해서 산책로 i 는 산책로 $i+1$ 보다 더 아름답다.
- P - 고급 식당이 위치하는 연못의 번호
- R - 산책로들을 표현한 2 차원 배열. 산책로 $i(0 \leq i < M-1)$ 는 연못 $R[i][0]$ 과 연못 $R[i][1]$ 을 연결한다. 한 산책로는 서로 다른 두개의 연못을 연결하며, 두 연못 사이에는 최대 하나의 산책로만 존재한다는 것에 주의하라.

- Q – 반의 수
- G – 각 반에 대한 K 의 값을 가지는 배열. 각 $i(0 \leq i < Q)$ 에 대해서 $G[i]$ 의 값은 반 i 가 목적지 P 에 도달하기 위해 이용하는 산책로의 개수인 K 의 값이다.

각 $i(0 \leq i < Q)$ 에 대해서 당신의 함수는 철수와 반 i 의 학생들이 정확히 $G[i]$ 개의 산책로를 이용한 후 연못 P 에 도달할 수 있는 모든 가능한 서로 다른 경로들의 갯수를 찾아야 한다. 각 반 i 에 대해서 당신의 함수는 반 i 의 경로의 수가 X 라는 의미로 $\text{answer}(X)$ 를 호출해야 한다. $\text{answer}()$ 함수가 호출되는 순서는 배열 G 에 주어진 순서와 동일해야 한다. 어떤 반에 대해 가능한 경로가 없는 경우는 $\text{answer}(0)$ 을 호출해야 한다.

예제

예제 1

그림 1에서 $N=6, M=6, P=0, Q=1, G[0]=3$ 인 경우이고, R 은 다음과 같다.

$R =$
1 2
0 1
0 3
3 4
4 5
1 5

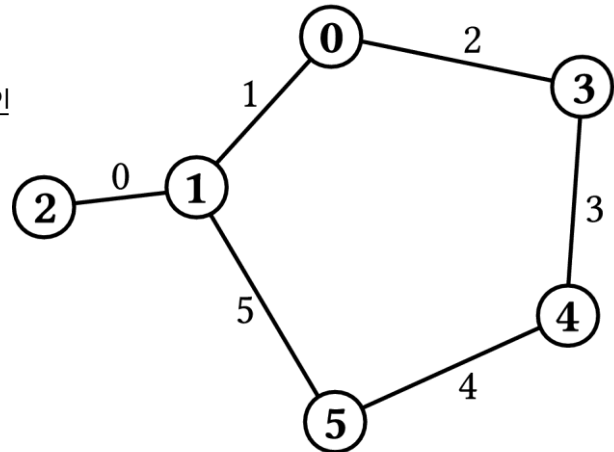


그림 1.

산책로의 번호가 작을 수록 더 아름답다는 것에 주의하자. 즉, 산책로 0이 가장 아름다운 것이고, 산책로 1이 그 다음으로 아름다운 것임을 알수 있다.

산책로의 개수가 3 이고 연못 0에서 끝나면서 이동 규칙을 만족하는 경로는 다음의 두가지 경우 밖에 없다.

- $1 \rightarrow 2 \rightarrow 1 \rightarrow 0$
- $5 \rightarrow 4 \rightarrow 3 \rightarrow 0$

첫번째 경로는 연못 1에서 시작한다. 그 곳에서 가장 아름다운 산책로를 택하면 연못 2로 가게 된다. 연못 2에서는 연못 1로 가는 방법 밖에는 없다. 이제 연못 1에서는 방금 사용한 산책로를 제외하면 연못 0으로 가는 산책로를 택하게 되어 목적지에 도착할 수 있다.

이 경우 당신의 함수는 $\text{answer}(2)$ 를 호출하여야 한다.

예제 2

그림 2에서 $N=5$, $M=5$, $P=2$, $Q=2$, $G[0]=3$, $G[1]=1$ 이고
 R 은 다음과 같다.

1 0
1 2
 $R=$ 3 2
1 3
4 2

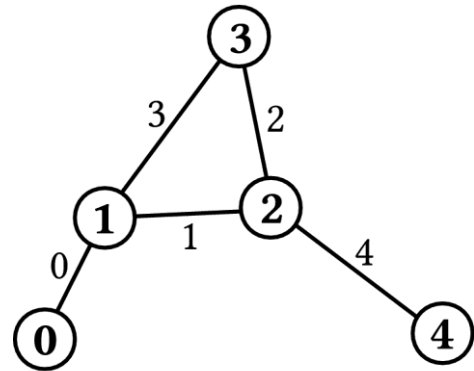


그림 2.

첫번째 반에 대해서 3개의 산책로를 이용하여 연못 2에 도착하는 방법은 하나 밖에 없으며 다음과 같다: $1 \rightarrow 0 \rightarrow 1 \rightarrow 2$.

두번째 반에 대해서 1개의 산책로를 이용하여 연못 2에 도착하는 방법은 다음의 두가지가 있다: $3 \rightarrow 2$ 와 $4 \rightarrow 2$.

따라서, 정확한 프로그램은 첫 번째 반에 대해 **answer(1)**을 호출하고 두 번째 반에 대해 **answer(2)**를 호출해야 하며 정확한 호출 순서를 지켜야 한다.

태스크

서브태스크 1 (49 점)

- $2 \leq N \leq 1\,000$
- $1 \leq M \leq 10\,000$
- $Q = 1$
- G 의 각 원소는 1 이상, 100 이하인 정수이다.

서브태스크 3 (31 점)

- $2 \leq N \leq 150\,000$
- $1 \leq M \leq 150\,000$
- $1 \leq Q \leq 2\,000$
- G 의 각 원소는 1 이상, 1,000,000,000 이하인 정수이다.

서브태스크 2 (20 점)

- $2 \leq N \leq 150\,000$
- $1 \leq M \leq 150\,000$
- $Q = 1$
- G 의 각 원소는 1 이상, 1,000,000,000 이하인 정수이다.

구현시 유의사항

제약조건

- CPU 시간 제한: 5 초
 - 메모리 제한: 256 MB
- 유의사항:** 스택 메모리 크기에 대한 제한은 명시되지 않는다. 스택 메모리는 전체 메모리 사용량에 포함된다.

인터페이스 (API)

- 프로그램 작업폴더: garden/
- 참가자가 작성할 파일: garden.c 또는 garden.cpp 또는 garden.pas
- 참가자 인터페이스: garden.h 또는 garden.pas
- 채점프로그램(grader) 인터페이스: gardenlib.h 또는 gardenlib.pas
- 견본 채점프로그램(sample grader): grader.c 또는 grader.cpp 또는 grader.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 P .
- 2 번째 줄부터 $M+1$ 번째 줄까지: 산책로들에 대한 기술이 주어진다; 즉, $i+2$ 번째 줄은 하나의 공백으로 분리된 $R[i][0]$ 과 $R[i][1]$ 가 있다 ($0 \leq i < M$).
- $M+2$ 번째 줄: Q .
- $M+3$ 번째 줄: 공백으로 분리된 정수들 수열의 배열 G .
- $M+4$ 번째 줄: 공백으로 분리된 정수들 수열로서, 예상한 해들의 배열.
- 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.)