문제3 : 열차의 이동

조차장은 열차의 객차들을 잇거나 떼어 내면서 열차를 유지하고 보수하는 장소이다. 조차장에는 열차를 움직일 수 있는 선로가 존재한다. 우리는 선로 상에서 열차의 객차들의 위치를 그래프로 나타낼 것이다. 그래프의 각 간선은 열차의 한 객차가 위치한 상태를 나타내고, 따라서 열차는 그래프상의 경로로 나타낸다. 이때, 경로상의 정점들과 간선들은 모두 달라야 한다. 특별히, 문제에서 주어지는 그래프는 항상 **트리**이다.

열차는 선로를 따라 이동시킬 수 있다. 열차의 이동은 단계 별로 이루어지고, 한 단계에 열차의 한쪽 끝 객차가 인접한 다른 간선으로 움직인다. 구체적으로, 열차를 나타내는 경로 P에 대해서, 한 단계 이동의 결과는 P의 한쪽 끝 정점과 인접한 P 밖의 간선 및 정점을 P에 추가하고, 다른 쪽 끝 정점과 간선을 P에서 제거하여 얻는 경로이다(그림1). 매 단계에서 경로의 길이는 변하지 않음에 주목하자.

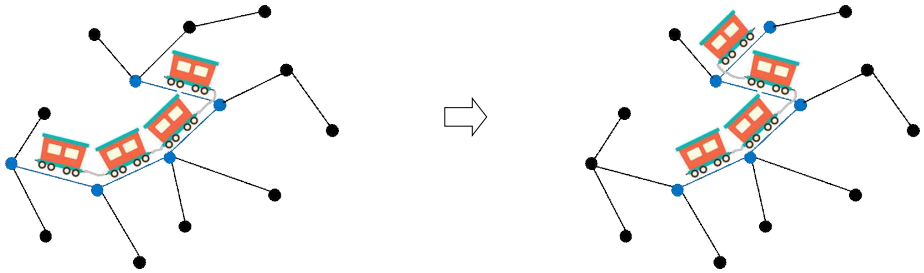
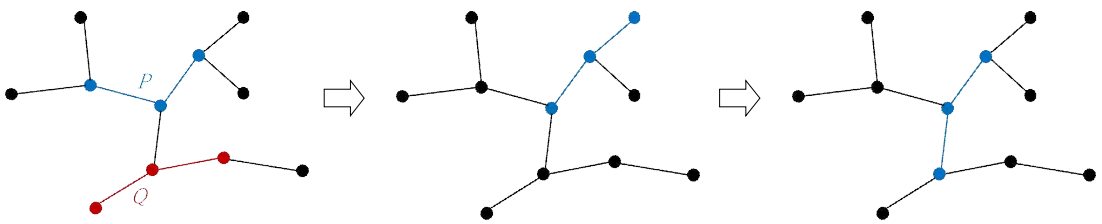


그림 1

각각 초기와 마지막 열차의 위치를 나타내는 길이 m의 경로 P와 Q가 주어진다. 경로의 길이는 경로에 포함된 간선의 개수이다. 여기서 두 경로 P와 Q는 어떠한 간선도 공유하지 않는다. 다시 말해서, P와 Q가 동시에 지나는 간선은 존재하지 않는다. 우리는 경로 P를 이동해서 마지막에 경로 Q가 되도록 해야 한다. 이때, 최소 단계의 이동을 찾아야 한다.

n개 정점을 가진 트리 T와 초기와 마지막 열차를 나타내는 길이 m의 경로 P와 Q가 주어질 때, P를 Q로 이동할 수 있는지 검사하고 이동할 수 있다면 최소 단계수를 출력하는 프로그램을 작성하시오.

예를 들어, 그림2에서 어떠한 간선도 공유하지 않는 길이 2의 두 경로 P와 Q가 주어진다. 그림에서와 같이 경로 P에서 Q로 5단계에 이동할 수 있고, 이것이 최소 단계의 이동이다.



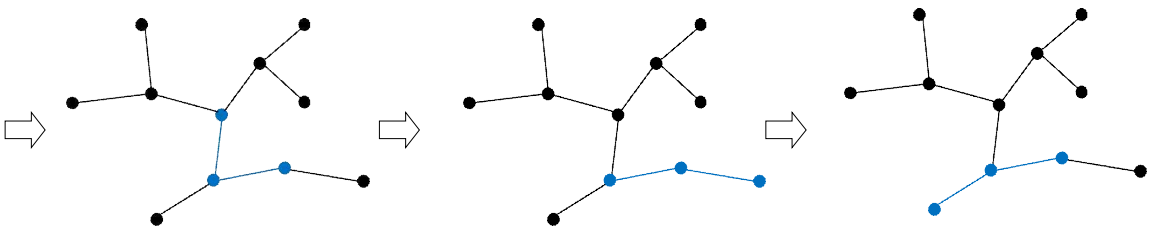


그림 2

여러분은 관리자를 위해 다음 두 가지 함수를 구현해야만 한다.

- `void init(int n, vector<int> X, vector<int> Y)` ; 최초에 호출되며 단 한번 호출되는 함수이다. 트리의 정점의 개수는 n이고 정점은 1부터 n까지의 정수로 나타낸다. X와 Y는 크기 n-1인 vector로 트리의 각 간선은 (X[i], Y[i])로 나타낸다.

- `long long train(vector<int> Z)` ; Z는 크기 4인 vector로 초기 경로 P의 두 끝점 Z[0]과 Z[1], 마지막 경로 Q의 두 끝점 Z[2]와 Z[3]를 나타낸다. P를 Q로 이동시키는 최소 단계수를 return한다. 만약, 이동할 수 없다면, -1을 return한다.

구현 세부사항

여러분은 train.cpp라는 이름의 정확히 하나의 파일을 제출해야만 한다. 이 파일에는 다음의 두 함수가 구현되어야 한다.

- `void init(int n, vector<int> X, vector<int> Y) ;`

- `long long train(vector<int> Z) ;`

이 함수는 위에서 설명한 것과 같이 동작하여야 한다. 물론, 다른 함수들을 만들어서 내부적으로 사용할 수 있다. 제출한 코드는 입출력을 수행하거나 다른 파일에 접근하여서는 안된다.

grader 예시

주어지는 grader는 다음과 같은 형식으로 입력을 읽는다:

- line 1: n q (n : 트리의 정점의 개수, $3 \le n \le 250{,}000$, 트리의 정점은 1부터 n으로 나타냄,
  q : 쿼리의 개수, $1 \le q \le 250{,}000$)
- 다음 n-1개의 줄 각각: x y (트리의 한 간선 (x, y), $1 \le x \neq y \le n$)
- 다음 q개의 줄 각각: a b c d (초기 경로 P의 두 끝점 a와 b, 마지막 경로 Q의 두 끝점 c와 d,
  쿼리로 주어지는 모든 경로 P의 길이의 합 $\le 1{,}000{,}000$)

주어지는 grader는 함수 `train`의 return 값을 출력한다. 만약, 이동할 수 없다면, -1을 출력한다.

서브태스크 1 [11 points]

- $n \le 80$, $q \le 80$.

서브태스크 2 [18 points]

- 경로 P의 길이 ≤ 2.

서브태스크 3 [34 points]

- 경로 P의 길이 ≤ 100, 모든 경로 P의 길이의 합 ≤ 250,000.

서브태스크 4 [36 points]

- n ≤ 1,000, q ≤ 1,000.

서브태스크 5 [51 points]

- 추가 제한이 없다.

[입력 예 1]

```
11 2
1 2
3 2
4 3
4 5
6 5
8 4
7 8
8 9
10 9
11 10
3 5 7 4
3 6 7 10
```

[출력 예 1]

```
3
7
```

아래는 예 1에 대해 함수 호출 및 그 결과를 차례대로 보여준다.

함수호출	결과 값
`init( 11, {1, 3, 4, 4, 6, 8, 7, 8, 10, 11},` `{2, 2, 3, 5, 5, 4, 8, 9, 9, 10} )`	
`train( {3, 5, 7, 4} )`	3
`train( {3, 6, 7, 10} )`	7