3-2. 갈라파고스 여행

형섭이와 민석이는 지난 여행 때 핀치 새를 제대로 보지 못한 것에 아쉬워하여 이번에 다시 갈라파고스 제도로 떠난다. 이번에는 페리 코스도 더 늘었으니 카약은 타지 않고 페리 코스로만 섬을 이동할 것이다.

갈라파고스 제도에는 총 N개의 섬과 M개의 페리 코스, K종류의 핀치가 있다. 섬의 번호는 각각 $0, 1, \ldots, N-1$이고, 페리 코스의 번호는 각각 $0, 1, \ldots, M-1$이며, 핀치의 번호는 각각 $0, 1, \ldots, K-1$이다. i $(0 \le i \le N-1)$번 섬에는 오직 한 종류의 핀치 $F[i]$만 있다. j $(0 \le j \le M-1)$번 페리 코스는 중요도가 j이며 $S[j]$번 섬과 $E[j]$번 섬 사이를 왕복 운행한다. 페리 코스만을 이용하여 N개의 섬을 모두 드나들 수 있다.

모든 섬을 방문한다면 모든 종류의 핀치를 볼 수 있다. 하지만 올해는 극심한 기상이변으로 기상 상황이 좋지 않다면 중요도가 가장 낮은 몇 개의 페리 코스를 운영하지 않을 것이다.

형섭이는 A번 핀치를 보고 싶어하며 민석이는 B번 핀치를 보고 싶어한다. 형섭이는 A번 핀치가 있는 섬에서 여행을 시작하여 갈 수 있는 모든 섬을 방문한다. 민석이는 B번 핀치가 있는 섬에서 여행을 시작하여 갈 수 있는 모든 섬을 방문한다.

형섭이와 민석이는 여행 도중에 만나서 같이 여행을 끝내려고 한다. 페리 코스가 많이 폐쇄된다면 형섭이와 민석이가 여행 도중에 만나지 못할 것이다. 형섭이와 민석이가 각자 조건을 만족하는 어떤 섬에서 여행을 시작하더라도 서로 만나지 못하기 위해 폐쇄되어야 하는 최소 페리 코스의 수를 구하는 프로그램을 작성하여라.

요구 사항

다음 함수를 구현해야 한다.

```
void Init(int K, int[] F, int[] S, int[] E)
```

- K: 핀치의 수.
- F: 길이 N의 배열. 모든 $0 \le i \le N-1$에 대해 $F[i]$는 i번 섬에 살고 있는 핀치의 번호이다.
- S: 길이 M의 배열. 모든 $0 \le j \le M-1$에 대해 $S[j]$는 j번 페리 코스의 시작 섬 번호이다.
- E: 길이 M의 배열. 모든 $0 \le j \le M-1$에 대해 $E[j]$는 j번 페리 코스의 도착 섬 번호이다.
- 이 함수에서는 `Separate` 함수에 필요한 전처리를 하면 된다.
- 이 함수는 테스트 케이스마다 `Separate` 함수가 호출되기 이전에 단 한 번 호출된다.

```
int Separate(int A, int B)
```

- A: 형섭이가 좋아하는 핀치의 번호.
- B: 민석이가 좋아하는 핀치의 번호.
- 이 함수는 형섭이가 A번 핀치가 있는 아무 섬에서 여행을 시작하고, 민석이가 B번 핀치가 있는 아무 섬에서 여행을 시작하더라도, 두 사람이 서로 만나지 못하기 위해 폐쇄되어야 하는 최소 페리 코스의 수를 반환한다. 페리 코스는 중요도가 낮은 순서대로 폐쇄됨에 유의하라.
- 이 함수는 `Init` 함수가 호출된 이후 Q번 호출된다.

제한

- $2 \le N \le 100\,000$
- $N - 1 \le M \le 300\,000$
- $2 \le K \le N$
- $1 \le Q \le 50\,000$
- 모든 $0 \le i \le N - 1$에 대해, $0 \le F[i] \le K - 1$
- 모든 $0 \le j \le M - 1$에 대해,
  - $0 \le S[j], E[j] \le N - 1$
  - $S[j] \neq E[j]$
- 모든 $0 \le f \le K - 1$에 대해, $F[i] = f$를 만족하는 $0 \le i \le N - 1$이 존재한다.
- 임의의 두 섬 사이를 왕복 운행하는 페리 코스는 최대 1개이다.
- 페리 코스만을 이용하여 N개의 섬을 모두 드나들 수 있다.
- 모든 `Separate` 함수 호출에 대해,
  - $0 \le A, B \le K - 1$
  - $A \neq B$

부분문제

1. (31점) $K = N$.
2. (39점) $K \le 1\,000$.
3. (30점) 추가 제약 조건은 없다.

예제

다음 호출을 고려해보자:

```
Init(5, [0, 1, 2, 0, 1], [0, 2, 2, 4, 1, 0], [1, 3, 4, 3, 2, 4])
Separate(0, 1)
Separate(1, 2)
Separate(0, 2)
```

`Separate` 함수의 정답은 각각 `6`, `5`, `3`이다.

샘플 인터페이스

문제 페이지에서 샘플 코드를 다운로드받을 수 있다. 만약 Visual Studio나 Eclipse, Code::Blocks 와 같은 IDE 툴을 사용한다면 `island.cpp`, `island.h`, `grader.cpp`를 한 프로젝트에 넣어서 컴파일하면 된다. 터미널에서 코드를 컴파일한다면 대회 페이지에 있는 컴파일 명령어를 이용하면 된다.

답안을 제출할 때에는 `island.cpp`를 제출하면 된다.

입력 형식

- line 1: N M K Q
- line 2: $F[i]$ ($0 \le i \le N-1$)
- line $3+j$ ($0 \le j \le M-1$): $S[j]$ $E[j]$
- line $3+M+k$ ($0 \le k \le Q-1$): $A[k]$ $B[k]$

여기서, $A[k]$와 $B[k]$는 k번 `Separate` 호출에 주어지는 값이다.

출력 형식

- line $1+k$ ($0 \le k \le Q-1$): k번 `Separate` 호출의 반환 값.