

멧돼지

사냥꾼 비룡이는 인하마을의 의뢰를 받아, 논과 밭을 헤집고 먹이를 탐색하는 흥포한 멧돼지를 잡으러 나섰다. 멧돼지 사냥 계획을 세우기 위해 마을 사람들에게 멧돼지에 대해 물어보았고, 다음의 습성들을 알아낼 수 있었다.

- 마을 근처에는 먹이가 있을 만한 여러 지역이 있고 몇몇 지역들 사이에는 두 지역을 직접 연결하는 길목이 있다. 직접 연결된 길목이 있는(인접한) 지역들이 아니더라도, 여러 지역을 경유하여 왕래가 가능할 수도 있지만, 다른 지역을 경유하더라도 서로 연결되는 길이 없고 숲으로 막혀 있는 지역들도 있을 수 있다.
- 멧돼지는 모든 지역들에 대해 먹이를 탐색한다. 단, 여러 지역들 중 고도가 가장 낮은 지역에서 먼저 먹이를 탐색하고, 연결된 길목을 따라 그 지역에 인접한 지역들을 방문하여 탐색한다. 단, 이미 탐색한 지역은, 다시 도착하더라도 재탐색을 하지는 않는다.
- 멧돼지가 특정 지역 A에서 먹이를 탐색한 후 다른 지역을 탐색하려고 할 때, 아직 탐색하지 않았고 지역 A에 인접한 지역들이 여러 곳 있다면, 그 중에서 고도가 가장 낮은 지역을 먼저 탐색한다.
- 탐색한 지역 B에서 더 이상 새롭게 탐색할 수 있는 지역이 없는 경우, B를 처음 탐색하러 올 때 들어왔던 길을 따라 이전 지역 A로 되돌아간다. 이후 A에 인접한 지역들 중 아직 탐색하지 않은 지역들이 있다면 위의 습성에 따라 계속해서 탐색한다.
- 아직 모든 지역을 탐색하지는 않았으나 길을 따라 탐색 가능한 지역은 없을 경우, 숲을 지나서 탐색을 계속한다. 이 때, 길이 없는 숲을 통과하여 도착한 지역은 아직 탐색되지 않은 지역들 중, 고도가 가장 낮은 지역이다. 이후 위 습성에 따라 탐색을 계속하고, 모든 지역들을 탐색하였다면 멧돼지는 먹이 탐색을 종료한다.

사냥꾼 비룡이는 인하마을의 촌장으로부터 마을 주변의 지도를 받아 멧돼지를 사냥하고자 한다. 지도에는 먹이가 있을 만한 지역들의 정보가 표시되어 있다. 멧돼지가 먹이를 찾기 위해 탐색한 지역 순서를 알아내는 프로그램을 작성하시오.

※ 프로그램의 실행 시간은 1 초, 메모리 사용량은 512MB 를 초과할 수 없다.

사용할 수 있는 언어는 C, C++로 제한한다. C++의 경우 main 함수 내의 시작 지점에 다음 내용을 추가함으로써 cin, cout 의 입출력 속도를 개선할 수 있다.

```
ios_base::sync_with_stdio(false);
cin.tie(NULL);
cout.tie(NULL);
```

단, 위의 내용을 추가할 경우 cin, cout 만 사용해야 하며, scanf, printf 등 C 입출력을 혼용해서 사용하면 안된다. C++의 std::endl 의 경우 출력 속도가 느리므로, cout<<endl; 대신 cout<<"\n";을 사용하는 것을 권장한다.

입력

첫 번째 줄에는 테스트 케이스 수 T ($1 \leq T \leq 10$)가 주어진다.

이후 각 테스트 케이스의 정보가 다음과 같이 주어진다.

- 첫 번째 줄에는 먹이가 있을 만한 지역의 수 N ($3 \leq N \leq 1,000$)과 지역 간 길목의 수 M ($3 \leq M \leq 100,000$)이 공백을 사이에 두고 주어진다.
- 두 번째 줄에는 지역 i ($0 \leq i < N$)의 고도, 즉 H_0 부터 H_{N-1} 이 공백을 사이에 두고 차례로 주어진다.
각 지역의 고도는 1이상 50,000이하의 정수이며, 모두 다르게 주어진다.
- 이후 M 개의 줄에, 길목이 존재하는 지역 번호 i, j ($i \neq j, 0 \leq i, j < N$)가 공백을 사이에 두고 한 줄에 하나씩 주어진다.

출력

테스트 케이스마다 맷돼지가 먹이를 찾기 위해 탐색한 지역순으로 지역 번호를 한 줄에 하나씩 출력한다.

예제 입출력

예제 입력	예제 출력
2	3
6 5	5
30 10050 456 25 70 63	4
0 2	0
4 5	2
2 1	1
3 5	3
4 3	0
5 5	4
30 10050 456 25 70	1
0 4	2
1 4	
2 0	
2 1	
1 0	