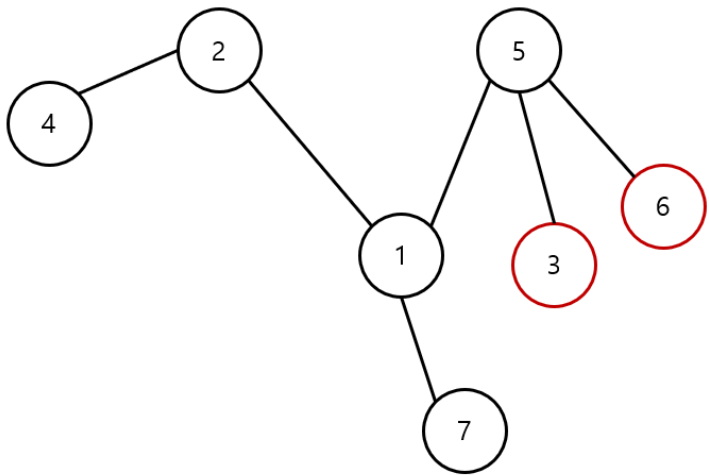


생태계 보호

동물학자 인하는 여러 지역의 생태계를 관찰하며 연구를 하고 있다. 각 지역은 1번부터 n 번까지의 번호가 부여되어 있다. 1번 지역은 모든 종이 서식하는 지역으로 가장 면적이 큰 지역이다. 각 지역들은 인접한 다른 지역과 다리로 연결되어 있으며, 임의의 두 지역을 연결하는 다리는 단 하나만 존재한다.

각 지역의 동물들을 연구하던 인하는 어느 날 일부 지역에서 신종 바이러스를 발견하였다. 해당 바이러스는 다행히 인간에게는 아무런 영향을 미치지 못하지만, 다른 동물들이 감염되면 즉시 죽게 된다. 또한 이 바이러스는 전염성이 매우 높기 때문에 한 지역에서 한 마리의 동물만 감염되어도 순식간에 해당 지역 내의 모든 동물들이 감염되어 죽게 된다. 또한, 일정 시간이 지나면 인접한 지역(다리로 연결된 지역)의 동물들도 감염시키게 된다. 모든 동물들을 살릴 수 없다고 판단한 인하는 두 지역을 잇는 다리를 막아서 추가적인 감염을 막으려 한다. 하지만 시간이 부족하기 때문에 하나의 다리만 막을 수 있다. 단, 모든 종이 서식하는 1번 지역은 반드시 바이러스로부터 감염을 막으면서 최대한 많은 지역의 감염을 막고자 한다.



예를 들어, 인하가 관찰하는 지역 생태계와 각 지역들을 잇는 다리들이 위 그림과 같고, 3번 지역과 6번 지역에서 동시에 바이러스가 발견되었다고 가정하자. 3번 지역과 5번 지역을 잇는 다리를 막으면 6번 지역과 5번 지역을 잇는 다리를 통해 모든 지역이 바이러스에 감염되지만, 1번 지역과 5번 지역을 잇는 다리를 막는다면, 5번 지역은 감염되지만 1, 2, 4, 7번 지역은 감염을 막을 수 있다.

인하가 관찰하는 지역 생태계와 각 지역들을 잇는 다리의 정보, 바이러스가 발견된 지역의 번호가 주어졌을 때, 1번 지역을 바이러스로부터 보호하면서 최대한 많은 지역을 보호하기 위해 막아야 하는 다리가 잇는 두 지역의 번호를 오름차순으로 출력한다. 단, 1번 지역을 보호할 수 없는 경우 -1을 출력한다.

※ 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 ($1 \leq T \leq 100$)가 주어진다.

이후 각 테스트 케이스의 정보가 다음과 같이 주어진다.

- 첫 번째 줄에 인하가 관찰하는 지역의 개수 n ($1 \leq n \leq 100,000$) 과 바이러스가 발견된 지역의 개수 m ($1 \leq m < n$)이 공백으로 구분되어 주어진다. 각 지역은 1번부터 n 번의 번호가 붙어 있다고 가정한다.
- 이후 $n - 1$ 개의 줄에 걸쳐, 두 지역을 잇는 다리의 정보 u, v ($1 \leq u, v \leq n, u \neq v$)가 주어진다. 이는 u 번 지역과 v 번 지역을 잇는 다리가 존재함을 의미한다. 다리의 정보는 중복되지 않는다.
- 이후 m 개의 바이러스가 발견된 지역의 번호가 한 줄에 공백으로 구분되어 주어진다. 이는 중복되지 않으며, 1번 지역은 주어지지 않는다.

출력

각 테스트 케이스마다 인하가 막아야 하는 다리가 연결하는 두 지역의 번호를 오름차순으로 공백으로 구분하여 한 줄에 출력한다.

예제 입출력

예제 입력	예제 출력
2	1 5
7 2	-1
1 2	
1 5	
1 7	
3 5	
5 6	
2 4	
3 6	
4 3	
1 2	
1 3	
1 4	
2 3 4	