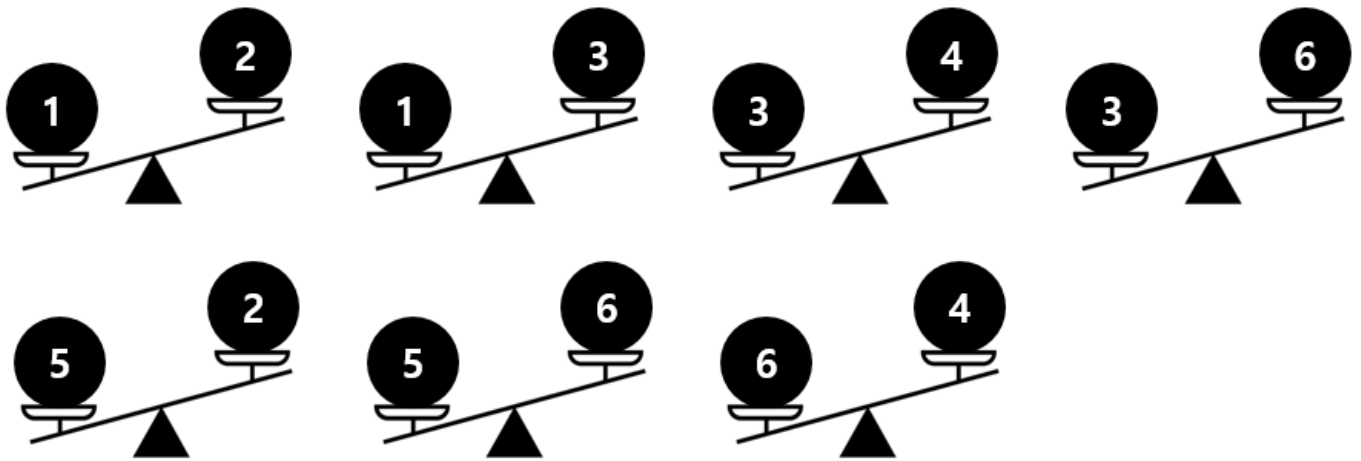


양팔 저울

박인하는 무게가 서로 다른 N 개의 공을 가지고 있다. 각 공들은 1번부터 N 번까지 번호가 적혀있으며, 크기와 모양이 같아 겉보기에는 어떤 공이 무거운지 전혀 알 수 없다. 박인하는 공들의 정확한 무게는 알고 있지 않지만, 특정 두 공의 무게를 양팔 저울을 통해 비교한 결과를 가지고 있다.

예를 들어, 공이 1번부터 6번까지 6개가 있고, 양팔 저울로 측정한 7개의 결과가 [그림 1]과 같이 주어진다 하자. [그림 1]의 측정결과를 잘 살펴보면, 1번 공과 4번 공은 무게를 직접적으로 비교하지 않았으나, 1번 공이 4번 공보다 더 무거운 것을 알 수 있다. 반면, 3번 공과 5번 공은 두 공 중 어떤 공이 무거운 공인지 결정할 수 없다.

박인하는 지금까지의 측정 결과를 바탕으로, 각 공들 보다 가볍다는 것이 결정된 공의 개수를 알고 싶어 한다. 위의 예시에서, 1번 공보다 가볍다는 것이 결정된 공은 2, 3, 4, 6번 공이며, 5번 공과의 비교 결과는 결정할 수 없으므로, 1번 공보다 가볍다는 것이 결정된 공의 개수는 4개이다.



[그림 1]

※ 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 줄에 테스트 케이스 수 T ($1 \leq T \leq 100$)가 주어진다.

각 테스트 케이스의 구성은 다음과 같다.

- 공의 개수 N ($2 \leq N \leq 100$)과 양팔 저울로 측정한 횟수 M ($1 \leq M \leq N(N - 1)/2$)이 한 줄에 주어진다.
- 이후 M 개의 줄에 걸쳐 공의 번호 A 와 B ($1 \leq A \neq B \leq N$)가 공백으로 구분되어 주어진다. 이는 A 번 공이 B 번 공보다 무거움을 의미한다.

단, 모순되는 입력은 주어지지 않는다. (즉, 공 A, B, C 에 대해, $A > B, B > C, C > A$ 인 입력은 주어지지 않는다.)

출력

각 테스트 케이스마다 1번부터 N 번 공에 대해, 각 공보다 확실히 가벼운 공의 개수를 공백으로 구분하여 한 줄로 출력한다.

예제 입출력

예제 입력	예제 출력
2	4 0 2 0 3 1
6 7	6 2 1 0 4 3 1
1 2	
1 3	
3 4	
3 6	
5 2	
5 6	
6 4	
7 9	
1 2	
1 5	
2 3	
5 3	
5 6	
6 3	
3 4	
6 7	
7 4	