

순열 게임

앨리스와 밥은 어릴때부터 친구이고, 머리를 쓰는 게임을 하는 것을 좋아한다. 오늘, 이 둘은 그래프에 관한 새로운 게임을 하고 있다.

이 게임은 0 부터 $m - 1$ 까지 번호가 매겨진 m 개의 정점과 0 부터 $e - 1$ 까지 번호가 매겨진 e 개의 간선을 가진 **연결** 그래프를 가지고 한다. 간선 i 는 정점 $u[i]$ 과 정점 $v[i]$ 을 연결한다.

또한 이 게임에서는 길이 n 인 순열 $p[0], p[1], \dots, p[n - 1]$ 을 사용하는데, $m \leq n$ 이다. 순열은 0 부터 $n - 1$ 까지의 수가 어떤 순서에 따라 정확하게 한번만 나오는 배열이다. 순열 p 의 **점수**는 $p[i] = i$ 인 인덱스 i 의 개수이다.

이 게임은 최대 10^{100} 턴 동안 지속된다. 각 턴마다, 다음과 같은 일이 벌어진다.

1. 만약 앨리스가 게임을 그만하기로 하면, 게임은 중단된다.
2. 그렇지 않으면, 앨리스는 **서로 다른 인덱스** $t[0], t[1], \dots, t[m - 1]$ ($0 \leq t[i] < n$)를 고른다. $t[0] < t[1] < \dots < t[m - 1]$ 일 **필요는 없다**는데 유의하자.
3. 밥은 0 이상 e 미만인 인덱스 j 를 고르고, $p[t[u[j]]]$ 와 $p[t[v[j]]]$ 를 맞바꾼다.

앨리스는 순열의 최종 점수를 최대로 하고 싶어 하고, 반대로 밥은 순열의 최종 점수를 최소로 하고 싶어 한다.

당신이 할 일은 앨리스를 도와서 밥을 이기게 하는 것이다. 밥이 하는 선택은 그레이더가 시뮬레이션해준다.

최적 점수는 앨리스와 밥 모두 최선의 선택을 하였을 때 최종적으로 얻게 되는 점수로 정의된다.

당신은 순열의 최적 점수를 알아내야 하며, 밥과 이 게임을 했을 때 진행되는 턴 중에서 최적 점수를 **적어도 한 번** 얻어야 한다.

앨리스의 전략은 밥이 최적인 행동을 하지 않는 경우를 포함해서 어떤 경우여라도 통해야 함에 유의하라.

Implementation details

다음 함수를 구현해야 한다.

```
int Alice(int m, int e, std::vector<int> u, std::vector<int> v,
          int n, std::vector<int> p)
```

- m : 그래프의 정점의 개수
- e : 그래프의 간선의 개수
- u, v : 그래프의 간선을 표현하는 길이 e 인 배열

- n : 순열의 길이
- p : 순열을 나타내는 길이 n 인 배열
- 이 함수는 정확히 한 번 호출된다.
- 이 함수는 정수를 리턴해야 하는데, 이 값은 이 게임의 최적 점수이다.

이 함수 내부에서, 다음 함수를 호출할 수 있다.

```
int Bob(std::vector<int> t)
```

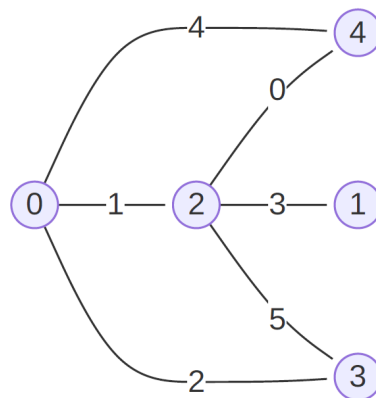
- t : 서로 다른 인덱스를 포함하는 길이 m 인 배열로 $0 \leq t[i] < n$ 이고 $i \neq j$ 이면 항상 $t[i] \neq t[j]$ 가 성립한다.
- 이 함수는 $0 \leq j < e$ 인 정수 j 를 리턴한다.
- 이 함수는 여러번 호출될 수 있다.

Example

다음 호출을 생각해보자.

```
Alice(5, 6, [4, 0, 3, 1, 4, 2], [2, 2, 0, 2, 0, 3],
      10, [8, 2, 7, 6, 1, 5, 0, 9, 3, 4])
```

그래프는 다음과 같은 모양이다.



p 는 처음에 $[8, 2, 7, 6, 1, 5, 0, 9, 3, 4]$ 이다.

위 제약조건에서, 이 순열의 최적 점수는 1이라는 것을 보일 수 있다.

앨리스가 다음과 같이 4번 플레이한다고 하자.

Bob에 전달된 파라미터 t	Bob의 리턴값	해당하는 p 의 인덱스들	Bob이 맞바꾼 다음 p
[3, 1, 5, 2, 0]	5	5, 2	[8, 2, 5, 6, 1, 7, 0, 9, 3, 4]
[9, 3, 7, 2, 1]	0	1, 7	[8, 9, 5, 6, 1, 7, 0, 2, 3, 4]
[5, 6, 7, 8, 9]	1	5, 7	[8, 9, 5, 6, 1, 2, 0, 7, 3, 4]
[7, 5, 2, 3, 6]	3	5, 2	[8, 9, 2, 6, 1, 5, 0, 7, 3, 4]

위에서 보인 선택은 설명을 편하게 하기 위해서이며, 앨리스와 밥의 선택이 최적일 수도 있음에 유의하자. 또한 앨리스는 게임을 바로 종료할 수 있다는데 주의하자. 처음 순열의 점수가 이미 최적 점수인 1이기 때문이다.

앨리스가 위와 같이 선택을 하고 나면, 순열의 점수는 3이다 ($p[2] = 2, p[5] = 5, p[7] = 7$).

마지막으로, 함수 Alice의 리턴값은 1이며, 이 순열의 최적 점수이다.

앨리스가 밥과 게임을 해서 3점을 얻었다고 하더라도, Alice의 리턴값이 1이 아니고 3이면 0점을 받는다는데 주의하시오.

Constraints

- $2 \leq m \leq 400$
- $m - 1 \leq e \leq 400$
- $0 \leq u[i], v[i] < m$
- $m \leq n \leq 400$
- $0 \leq p[i] < n$
- 주어진 그래프는 연결 그래프이며, 간선의 양 끝점은 반드시 서로 다르다. 또한, 두 정점을 잇는 에지는 최대 하나이다.
- p 는 순열이다. 즉, $i \neq j$ 이면 $p[i] \neq p[j]$ 이다.

Subtasks

1. (6 points) $m = 2$
2. (6 points) $e > m$
3. (10 points) $e = m - 1$
4. (24 points) $e = m = 3$
5. (24 points) $e = m = 4$
6. (30 points) $e = m$

각 서브태스크마다 부분 점수를 받을 수 있다. k 가 턴 수일 때, (즉, Bob () 을 호출한 수) r 이 한 서브태스크의 모든 테스트케이스에서 $\frac{k}{n}$ 의 최대값이라고 하자. 그러면, 이 서브태스크에 대한 점수에 다음 수가 곱해진다.

조건	곱하는 수
$12 \leq r$	0
$3 < r < 12$	$1 - \log_{10}(r - 2)$
$r \leq 3$	1

특히, $3n$ 턴 이내에 문제를 푼다면 이 서브태스크에 대해서 만점을 받을 수 있다. $12n$ 턴 이상이 걸린다면 이 서브태스크에 대해서 0점을 받게 된다. (Output isn't correct로 표시).

Sample Grader

샘플 그레이더는 다음 양식으로 입력을 읽는다.

- line 1: $m\ e$
- line $2 + i$ ($0 \leq i \leq e - 1$): $u[i]\ v[i]$
- line $2 + e$: n
- line $3 + e$: $p[0]\ p[1]\ \dots\ p[n - 1]$

샘플 그레이더는 다음 양식으로 출력한다.

- line 1: 최종 순열 p
- line 2: `Alice()`의 리턴값
- line 3: 최종 순열의 점수
- line 4: 턴 수