


[카테고리 없음](#)

## [BOJ](Python) 백준 9466번: 팀 프로젝트

그리버 | 2023. 11. 28. 17:24 | 수정 | 삭제

[문제](#)  
[풀이](#)



### 신규 회원 모두 1만 코인 지급

타로,사주,신점은 홍카페

지금 알아보기

### 문제

<https://www.acmicpc.net/problem/9466>



#### 9466번: 팀 프로젝트

이번 가을학기에 '문제 해결' 강의를 신청한 학생들은 팀 프로젝트를 수행해야 한다. 프로젝트 팀원 수에는 제한이 없다. 심지어 모든 학생들이 동일한 팀의 팀원인 경우와 같이 한 팀만 있을

[www.acmicpc.net](http://www.acmicpc.net)

주제	방향 그래프
시간 / 메모리 제한	3초 / 256MB
정답 비율	24.043%

이번 가을학기에 '문제 해결' 강의를 신청한 학생들은 팀 프로젝트를 수행한다.

- 프로젝트 팀원 수에는 제한이 없다.
  - 모든 학생들이 동일한 팀의 팀원인 경우처럼 한 팀만 있을 수도 있다.

프로젝트 팀을 구성하기 위해 모든 학생들은 프로젝트를 함께하고 싶은 학생을 선택한다.

- 각자 단 한 명만 선택할 수 있다.
- 혼자 하고 싶은 학생은 자기 자신을 선택하는 것도 가능하다.

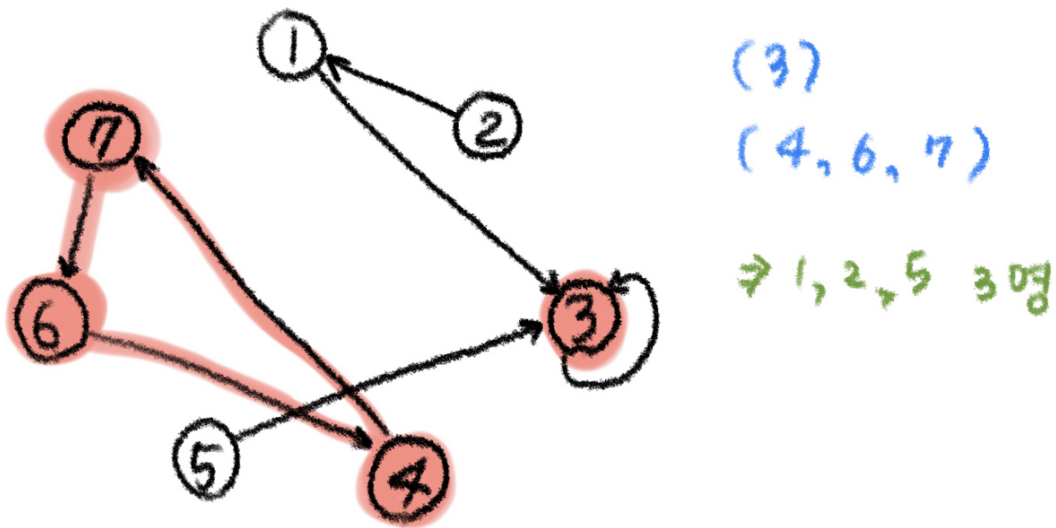
학생들이  $s_1, s_2, \dots, s_r$ 이라 할 때, 아래의 경우에만 한 팀이 될 수 있다.

- $r=1$ 이고  $s_1$ 이  $s_1$ 을 선택하는 경우
- $s_1$ 이  $s_2$ 를 선택하고,  $s_2$ 가  $s_3$ 를 선택하고, ...,  $s_{r-1}$ 이  $s_r$ 을 선택하고,  $s_r$ 이  $s_1$ 을 선택하는 경우

예를 들어, 한 반에 7명의 학생이 있고 아래와 같이 지목을 했을 경우를 생각해보자.

1	2	3	4	5	6	7
3	1	3	7	3	4	6

위의 경우는 아래의 그림과 같은 연결 상태가 된다.



이 경우 (3), (4, 6, 7)이 한 팀을 이루고 1, 2, 5의 경우 팀을 이루지 못한다.

주어진 선택의 결과를 보고 어느 프로젝트 팀에도 속하지 않는 학생들의 수를 계산하는 프로그램을 작성하라.

## 입력

첫째 줄에 테스트 케이스의 개수  $T$ 가 주어진다.

- 각 테스트 케이스의 첫 줄에는 학생의 수가 정수  $n$  ( $2 \leq n \leq 100,000$ )으로 주어진다.
- 각 테스트 케이스에 둘째 줄에는 선택된 학생들의 번호가 주어진다.

## 출력

각 테스트 케이스마다 한 줄에 출력하고,

각 줄에는 프로젝트 팀에 속하지 못한 학생들의 수를 나타내면 된다.

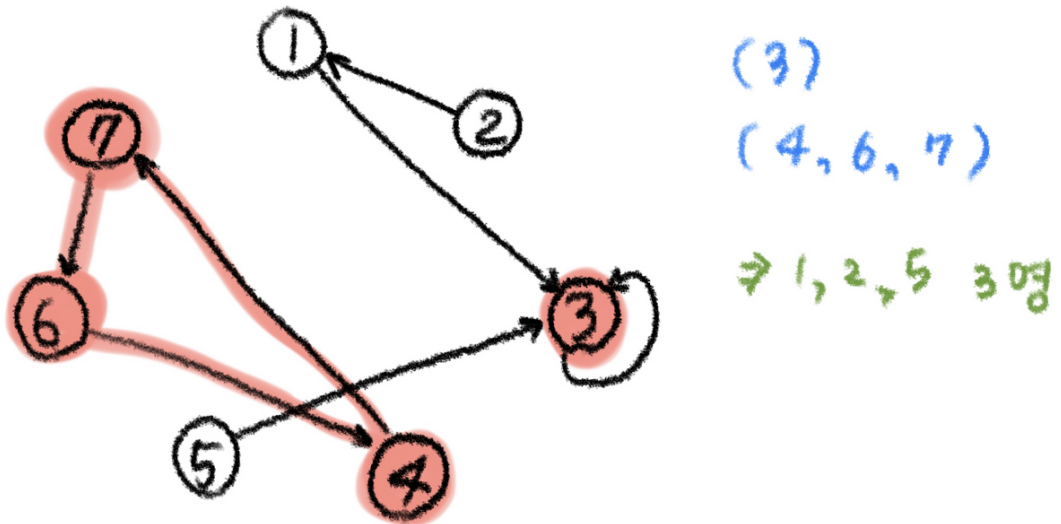
- 예제 입력 1

```

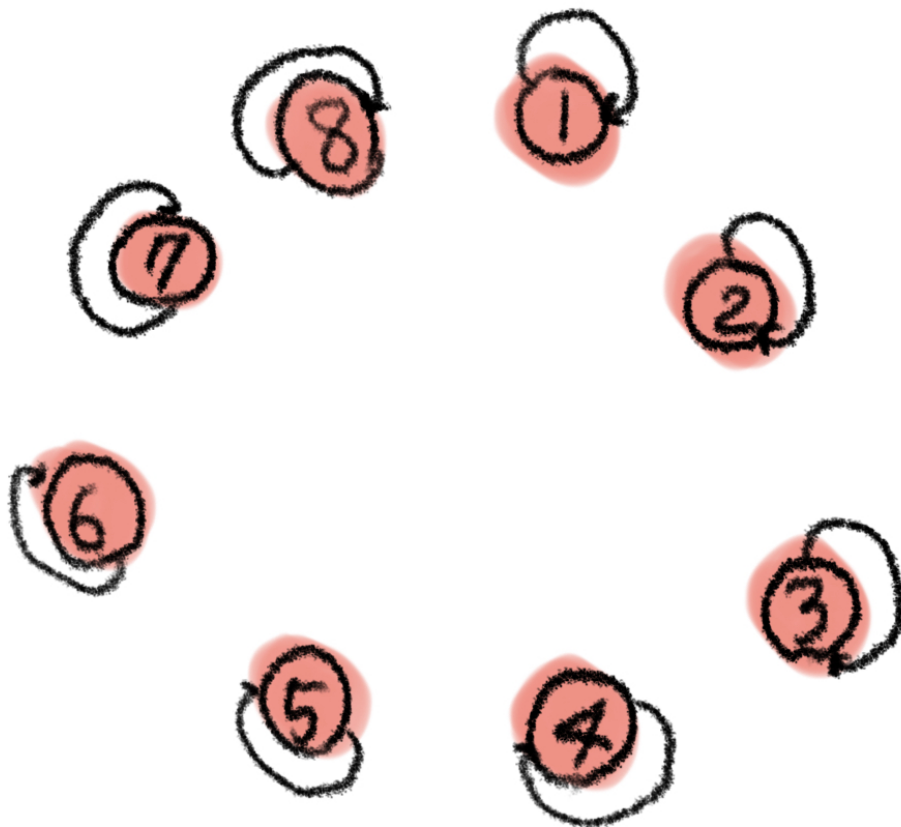
1 2
2 7
3 3 1 3 7 3 4 6
4 8
5 1 2 3 4 5 6 7 8
python
Copy

```

첫 번째 경우



두 번째 경우



- 예제 출력 1

```
1 3
2 0
```

python

## 풀이

이 문제는 그래프의 연결 관계를 보고 학생들의 팀 구성 상황에 대한 답을 내면 되는 문제이다.

그리고 각 학생이 다른 학생을 선택하는 것으로 방향이 생기게 되므로 특히 방향 그래프에 대한 문제이다.

방향 그래프를 그리며 이 방향 그래프의 연결 관계가 cycle을 이룬다면 한 팀으로 생각하면 된다.

cycle과 같은 그래프 용어에 대한 설명은 아래의 글을 참고하면 된다.

<https://classic-griver.tistory.com/185>



### [알고리즘](C언어) 그래프 알고리즘

1. 그래프 ADT 그래프 ADT 그래프(graph):  $G = (V, E)$  혹은  $G(V, E)$ 로 표시  $V$ : 정점(vertex)이라 불리는 노드의 집합  $E$ : 간선(edge)이라 불리는 정점쌍들의 집합 정점과 간선은 원소, 즉 정보를 저장 예 ...

classic-griver.tistory.com

결과적으로 방향 그래프 연결관계에서 cycle을 이루지 않는 학생의 수를 구하는 문제이다.

## 사이클(Cycle)

각 vertex가 사이클을 이루는지 이루지 않는지 판별하는 함수를 만든다면 이 문제를 풀 수 있다.

사이클은 경로를 따라 가다보면 결국 자기 자신으로 돌아오게 되는 특징을 가지고 있다.

모든 vertex를 순회하며,

- 해당 vertex에서 출발해 인접 vertex로 계속 이동했을 때 최종 vertex가 자기 자신이 되면 cycle,
- 더이상 인접 vertex가 존재하지 않거나 최종 vertex가 자기 자신이 아니면 bicycle로 판단한다.

이 특징을 이용해 제작한 함수는 아래와 같다.

각 학생은 단 한 명의 학생만 지목할 수 있기에 이렇게 코드를 작성할 수 있다.

```
Copy

1 import sys
2
3
4 class Graph:
```

```

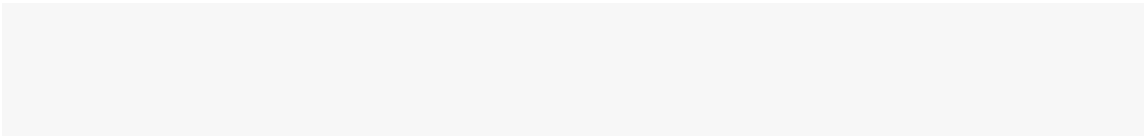
5
6     def __init__(self, vertex_num=None):
7         # 인접 리스트
8         self.adj_list = []
9         self.vtx_num = 0
10
11         self.vtx_arr = []
12
13         if vertex_num:
14             self.vtx_num = vertex_num
15             self.vtx_arr = [True for _ in range(self.vtx_num)]
16             self.adj_list = [[] for _ in range(self.vtx_num)]
17
18     def add_vertex(self):
19         self.adj_list.append([])
20         self.vtx_num += 1
21         self.vtx_arr.append(True)
22         return self.vtx_num - 1
23
24     def add_edge(self, u, v):
25         self.adj_list[u].append(v)
26
27
28 T = int(sys.stdin.readline().strip())
29
30 for _ in range(T):
31     n = int(sys.stdin.readline().strip())
32     num = 0
33     l = list(map(int, sys.stdin.readline().strip().split()))
34     g = Graph(n)
35     for i in range(n):
36         g.add_edge(i, l[i] - 1)
37
38     # cycle
39     for i in range(n):
40         cycle = 0
41         adj_vtx = g.adj_list[i][0]
42         for _ in range(n):
43             if adj_vtx == i:
44                 cycle = 1
45                 break
46             adj_vtx = g.adj_list[adj_vtx][0]
47         if cycle == 0:
48             num += 1
49     print(num)

```

python

그렇지만 이런 식으로 작성할 경우 시간 초과가 발생하였다.





☐ Secret

안녕하세요! 어떤 댓글이든 환영합니다! 📌

댓글달기