9466번: 텀 프로젝트

문제

이번 가을학기에 '문제 해결' 강의를 신청한 학생들은 텀 프로젝트를 수행해야 한다. 프로젝트 팀원 수에는 제한이 없다. 심지어 모든 학생들이 동일한 팀의 팀원인 경우와 같이 한 팀만 있을 수도 있다. 프로젝트 팀을 구성하기 위해, 모든 학생들은 프로젝트를 함께하고 싶은 학생을 선택해야 한다. (단, 단 한명만 선택할 수 있다.) 혼자 하고 싶어하는 학생은 자기 자신을 선택하는 것도 가능하다.

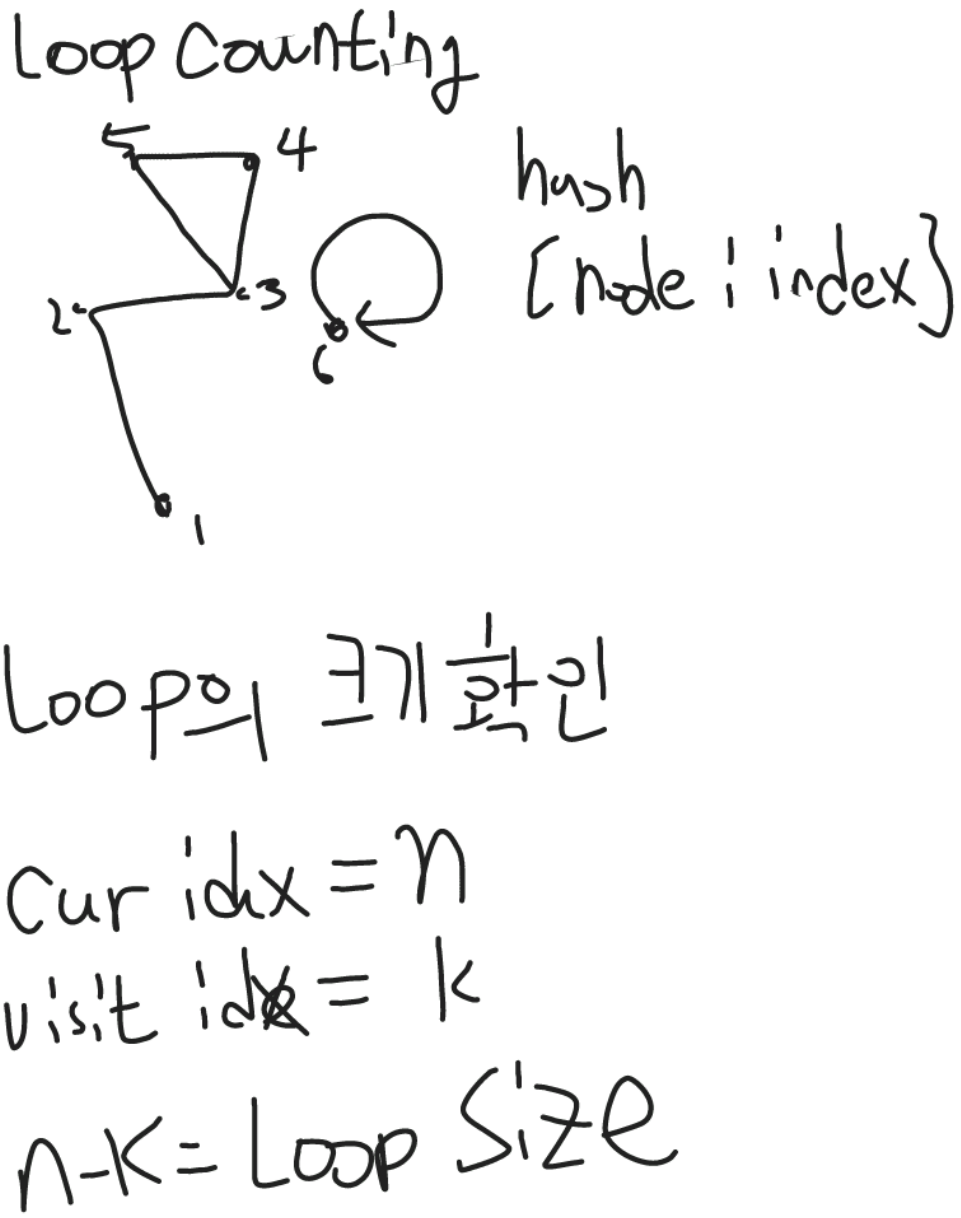
학생들이(s_1, s_2, ..., s_r)이라 할 때, r=1이고 s_1이 s_1을 선택하는 경우나, s_1이 s_2를 선택하고, s_2가 s_3를 선택하고,..., s_{r-1}이 s_r을 선택하고, s_r이 s_1을 선택하는 경우에만 한 팀이 될 수 있다.

예를 들어, 한 반에 7명의 학생이 있다고 하자. 학생들을 1번부터 7번으로 표현할 때, 선택의 결과는 다음과 같다.

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

위의 결과를 통해 (3)과 (4, 7, 6)이 팀을 이룰 수 있다. 1, 2, 5는 어느 팀에도 속하지 않는다.

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



문제 유형 : 그래프, 탐색

dfs나 bfs를 이용하여 그래프를 탐색하는 문제이다. 그래프를 탐색하며, loop를 감지하고 그 루프의 크기를 합산하여 출력하면 되는 문제이다. 루프를 감지하는데 있어서 여러가지 예외처리를 못하여 구현에 애를 먹었으며, 알고리즘을 떠올리고도, 어떻게 하면 구현할지 한참 ㅇ르 생각했다. 구현 능력과 집중력의 부족이다. 무엇인가를 만들 때 바싹 집중해야지 시간을 안 버린다. 생각하고 생각하자.

자바 코드

```
import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.HashMap;


public class Main {

public static BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
```

```
public static ArrayList<Integer>[] cnj;
public static int numOfS;
public static int visit[];
public static int cnt = 0;
public static HashMap<Integer,Integer> m = new HashMap<>();
public static void main(String args[])throws Exception {
int t = Integer.parseInt(in.readLine());
while(t!=0) {
numOfS=Integer.parseInt(in.readLine());
getInput();
for(int i=0; i<numOfS; i++) {
}
for(int i =0 ; i< numOfS; i++){
if(visit[i]==0)dfs(i,1,1);
}
System.out.println(numOfS-cnt);
cnt=0;
t--;
}
}
public static void getInput() throws Exception{
cnj = new ArrayList[numOfS];
visit = new int[numOfS];
for(int i =0; i< numOfS; i++) {
visit[i] = 0;
cnj[i] = new ArrayList<Integer>();
}
String line[] = in.readLine().split(" ");
for (int i =0 ; i< numOfS ; i++) {
int nex =Integer.parseInt(line[i])-1;
cnj[i].add(nex);
}
}
public static void dfs(int curNd, int dfsNum, int ndCnt){
if (visit[curNd]!=0){
if(visit[curNd]!= dfsNum)return;
if(m.containsKey(curNd)){
cnt += (ndCnt - m.get(curNd));
return;
```

```
}
}
else{
visit[curNd] = dfsNum;
m.put(curNd, ndCnt);
dfs(cnj[curNd].get(0),dfsNum,ndCnt+1);
m.clear();
}
}
}
```

python3

```
cnt = 0
m = dict()
stNum= 0
cnj = None
visit = None

def getInput():
  global cnj,visit ,stNum
  stNum= int(input())
  cnj = list();visit = list()
  for i in range (stNum):
  visit.append(0)
  cnj.append([])
  inputs = [int(i)-1 for i in input().split()]
  for i in range (stNum) :
  cnj[i].append(inputs[i])
def dfs(curNd,dfsN,ndCnt):
  global cnt
  if visit[curNd] !=0 :
  if visit[curNd]!=dfsN : return
  cnt += ndCnt - m[curNd]
  return
  visit[curNd] = dfsN
  m.setdefault(curNd,ndCnt)
  dfs(cnj[curNd][0],dfsN,ndCnt+1)
  m.clear()
```

```
if __name__ == "__main__" :
  t = int(input())
  while t :
  getInput()
  for i in range (stNum):
  if visit[i] == 0:
  dfs(i,i+1,1)
  print(stNum-cnt)
  cnt=0

  t-=1
```

입력

첫째 줄에 테스트 케이스의 개수 T가 주어진다. 각 테스트 케이스의 첫 줄에는 학생의 수가 정수 n (2 ≤ n ≤ 100,000)으로 주어진다. 각 테스트 케이스의 둘째 줄에는 선택된 학생들의 번호가 주어진다. (모든 학생들은 1부터 n까지 번호가 부여된다.)

출력

각 테스트 케이스마다 한 줄에 출력하고, 각 줄에는 프로젝트 팀에 속하지 못한 학생들의 수를 나타내면 된다.

예제 입력

```
2
7
3 1 3 7 3 4 6
8
1 2 3 4 5 6 7 8
```

예제 출력

```
3
0
```

힌트