

5567번: 결혼식

5567번 제출 맞은 사람 슛코딩 풀이 풀이 작성 풀이 요청 재채점/수정

문제 추천 채점 현황 강의 ▼

결혼식

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	128 MB	1302	551	481	44.210%

문제

상근이는 자신의 결혼식에 학교 동기 중 자신의 친구와 친구의 친구를 초대하기로 했다. 상근이의 동기는 모두 N명이고, 이 학생들의 학번은 모두 1부터 N까지이다. 상근이의 학번은 1이다.

상근이는 동기들의 친구 관계를 모두 조사한 리스트를 가지고 있다. 이 리스트를 바탕으로 결혼식에 초대할 사람의 수를 구하는 프로그램을 작성하시오.

문제 유형 : 그래프 , BFS flooding

접근 방법 : 데이터 들이 선형 데이터 (일방향적인 순서)가 아니고 서로를 가르키는 비선형 데이터 였다. 그리고 기준점으로 부터 몇 노드를 거쳐 도착 할 수 있는지 켜야 했다. BFS 탐색을 통해 노드간의 관계를 탐색하였다. 알고리즘은 금방 떠올렸으나, 구현하기 편한 DFS로만 해결 하다가 오랜만에 BFS를 사용 하니 구현할 때 버벅 거렸다.

BFS문제도 많이 풀어보자. BFS도 완전 탐색이다. Queue라는 자료구조를 사용해서 노드의 방문순서를 기억하고 조절 할 수 있다. 그러므로 최단 검색 및 노드의 방문 순서를 알 수 있다.

```
from collections import deque as queue
v =int(input()); e = int(input())
cnj =[[ ] for i in range(v)]
visit = dict()
q = queue()
for i in range (v) : visit.setdefault(i,0)
for i in range(e) :
```

```

cur , nex = map(lambda x : int(x)-1 ,input().split())
cnj[cur].append(nex)
cnj[nex].append(cur)
def bfs(cur,idx) :
    global visit
    q.append(cur)
    visit[cur] =idx
    cnt = idx
    while(q):
        nex= q.popleft()
        cnt = visit[nex]+1
        for i in cnj[nex] :
            if visit[i] == 0:
                q.append(i)
                visit[i] = cnt
bfs(0,1)
#print(cnj)
#print(visit)
a = list(map(lambda x:1 if x>1 and x<4 else 0 ,visit.values()))
print(sum(a))

```

입력

첫째 줄에 상근이의 동기의 수 n ($2 \leq n \leq 500$)이 주어진다. 둘째 줄에는 리스트의 길이 m ($1 \leq m \leq 10000$)이 주어진다. 다음 줄부터 m 개 줄에는 친구 관계 a_i b_i 가 주어진다. ($1 \leq a_i < b_i \leq n$) a_i 와 b_i 가 친구라는 뜻이며, b_i 와 a_i 도 친구관계이다.

출력

첫째 줄에 상근이의 결혼식에 초대하는 동기의 수를 출력한다.

예제 입력

```

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

```

예제 출력

```

3

```

힌트

2와 3은 상근이의 친구이다. 또, 3과 4는 친구이기 때문에, 4는 상근이의 친구의 친구이다. 5와 6은 친구도 아니고, 친구의 친구도 아니다. 따라서 2,3,4 3명의 친구를 결혼식에 초대한다.

출처

[Olympiad](#) > [일본정보올림피아드 예선](#) > [JOI 2010 예선 3번](#)

- 문제를 번역한 사람: baekjoon
- 문제의 오타를 찾은 사람: etaehyun4