

사람들을 조사해 서로 연결되었는지 확인한다. (한 단계씩 탐색)

창용 마을에는 N명의 사람이 살고 있다. 사람은 편의상 1번부터 N번 사람까지 번호가 붙어 있다고 가정한다. 두 사람은 서로 알고 있는 관계일 수 있고, 아닐 수 있다. 두 사람이 서로 아는 관계이거나 몇 사람을 거쳐서 알 수 있는 관계라면, 이러한 사람들을 모두 다 묶어서 하나의 무리라고 한다. 창용 마을에 몇 개의 무리가 존재하는지 계산하는 프로그램을 작성하라.

BFS 활용!

[입력]

첫 번째 줄에 테스트 케이스의 수 T가 주어진다.

각 테스트 케이스의 첫 번째 줄에는 각각 창용 마을에 사는 사람의 수와 서로를 알고 있는 사람의 관계 수를 나타내는 두 정수 N, M($1 \leq N \leq 100$, $0 \leq M \leq N(N-1)/2$) 이 공백 하나로 구분되어 주어진다.

이후 M개의 줄에 걸쳐서 서로를 알고 있는 두 사람의 번호가 주어진다. 같은 관계는 반복해서 주어지지 않는다.

[출력]

각 테스트 케이스마다 '#x'(x는 테스트케이스 번호를 의미하며 1부터 시작한다)를 출력하고, 무리의 개수를 출력한다.

예시) N = 6 (정점의 수)
M = 5 (간선의 수)

arr = [(1,2), (2,5), (5,1), (3,4), (4,6)]



i) 인접배열 생성 (관계 정리)

ii) 방문리스트 생성 (반복 제거를 위함)

iii) Queue를 활용하여 인접한 정점이고 이전에 방문하지 않은 정점 방문

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

• 무리의 수를 구해야 하기 때문에 방문하지 않은 임의의 정점을 무리의 시작점으로 간주한다. → 이 때 무리로 카운팅!

• 이 시작점을 시작으로 인접정점들을 반복하여 방문! 방문이 끝나면 다른 방문하지 않은 임의의 정점 찾기 (반복)

• 단, 혼자 있는 사람도 무리로 간주한다

[BOJ] 2667 단지번호 붙이기

<그림 1>과 같이 정사각형 모양의 지도가 있다. 1은 집이 있는 곳을, 0은 집이 없는 곳을 나타낸다. 철수는 이 지도를 가지고 연결된 집의 모임인 단지를 정의하고, 단지에 번호를 붙이려 한다. 여기서 연결되었다는 것은 어떤 집이 좌우, 혹은 아래위로 다른 집이 있는 경우를 말한다. 대각선상에 집이 있는 경우는 연결된 것이 아니다. <그림 2>는 <그림 1>을 단지별로 번호를 붙인 것이다. 지도를 입력하여 단지수를 출력하고, 각 단지에 속하는 집의 수를 오름차순으로 정렬하여 출력하는 프로그램을 작성하시오.

[입력]

첫 번째 줄에는 지도의 크기 N(정사각형이므로 가로와 세로의 크기는 같으며 $5 \leq N \leq 25$)이 입력되고, 그 다음 N줄에는 각각 N개의 자료(0 혹은 1)가 입력된다.

[출력]

첫 번째 줄에는 총 단지수를 출력하시오. 그리고 각 단지내 집의 수를 오름차순으로 정렬하여 한 줄에 하나씩 출력하시오.

0	1	1	0	1	0	0
0	1	1	0	1	0	1
1	1	1	0	1	0	1
0	0	0	0	1	1	1
0	1	0	0	0	0	0
0	1	1	1	1	1	0
0	1	1	1	0	0	0

<그림 1>

0	1	1	0	2	0	0
0	1	1	0	2	0	2
1	1	1	0	2	0	2
0	0	0	0	2	2	2
0	3	0	0	0	0	0
0	3	3	3	3	3	0
0	3	3	3	0	0	0

<그림 2>

• 2차배열을 순회하면서 방문한 적 없는 집을 찾는다 → 단지의 수 기록!

↳ 방문한 적 있는지 체크하는 배열 생성

• 그 집을 시작점으로 주변의 집을 탐색한다. (방문한 적 없는)

↳ 인접배열 생성 (delta 활용)

• 탐색시, 방문여부와 단지내 집의 수를 기록

↳ 변수 생성

N^2 개의 방이 $N \times N$ 형태로 늘어서 있다. 위에서 i 번째 줄의 왼쪽에서 j 번째 방에는 1이상 N^2 이하의 수 $A_{i,j}$ 가 적혀 있으며, 이 숫자는 모든 방에 대해 서로 다르다.당신이 어떤 방에 있다면, 상하좌우에 있는 다른 방으로 이동할 수 있다. 물론 이동하려는 방이 존재해야 하고, 이동하려는 방에 적힌 숫자가 현재 방에 적힌 숫자보다 정확히 1 더 커야 한다.처음 어떤 수가 적힌 방에서 있어야 가장 많은 개수의 방을 이동할 수 있는지 구하는 프로그램을 작성하라.

[입력]

첫 번째 줄에 테스트 케이스의 수 T 가 주어진다. 각 테스트 케이스의 첫 번째 줄에는 하나의 정수 N ($1 \leq N \leq 103$)이 주어진다. 다음 N 개의 줄에는 i 번째 줄에는 N 개의 정수 $A_{i,1}, \dots, A_{i,N}$ ($1 \leq A_{i,j} \leq N^2$) 이 공백 하나로 구분되어 주어진다. $A_{i,j}$ 는 모두 서로 다른 수이다.

[출력]

각 테스트 케이스마다 '#x'(x는 테스트케이스 번호를 의미하며 1부터 시작한다)를 출력하고, 한 칸을 띄운 후, 처음에 출발해야 하는 방 번호와 최대 몇 개의 방을 이동할 수 있는지를 공백으로 구분하여 출력한다.이동할 수 있는 방의 개수가 최대인 방이 여럿이라면 그 중에서 적힌 수가 가장 작은 것을 출력한다.

예시) $N=3$

arr=[(9,3,4),(6,1,5),(7,8,2)]

9	3	4
6	1	5
7	8	2

- 배열을 순회하면서 delta로 주변 Check!
& 나보다 1이 큰 숫자가 있으면 기록한다
& 나보다 1이 작은 숫자가 있으면 기록한다

lst =

	(0,2)	(0,1)
(2,0)		(0,2)
(1,0)	(2,1)	

- 방문을 Check 하는 빈배열
- lst 배열 순회하여 탐색 시작 정점 찾기
조건: 값의 길이가 1 & 방문한 적이 없음
시작점 아 도착점이니가!

visit =

0	0	0
0	0	0
0	0	0

- 탐색 시작 정점 기준으로 방문하지 않고 1차이가 나는 정점 탐색 & 방문기록

➡ 코드를 만들고 보니 너무복잡함... 수정!

i) 인접배열 없이 그냥 탐색한다
(인접배열을 통한 탐색이 아닌 delta를 활용)

ii) 결과출력을 용이하게 하기 위해 탐색정점을 따로 기록한다.

```
for i → N
  for j → N
    if visit[i][j] == 0:
      i) 탐색하기 (queue와 delta 활용)
      ii) 탐색 완료 후, 결과 도출을 위한 비교
        조건) 방의 개수가 더 많다
        조건) 방의 개수가 같다면 방번호가 더 작다
```


A1, A2, ... , AN의 N개의 자연수가 주어졌을 때, 최소 1개 이상의 수를 선택하여 그 합이 K가 되는 경우의 수를 구하는 프로그램을 작성하시오.

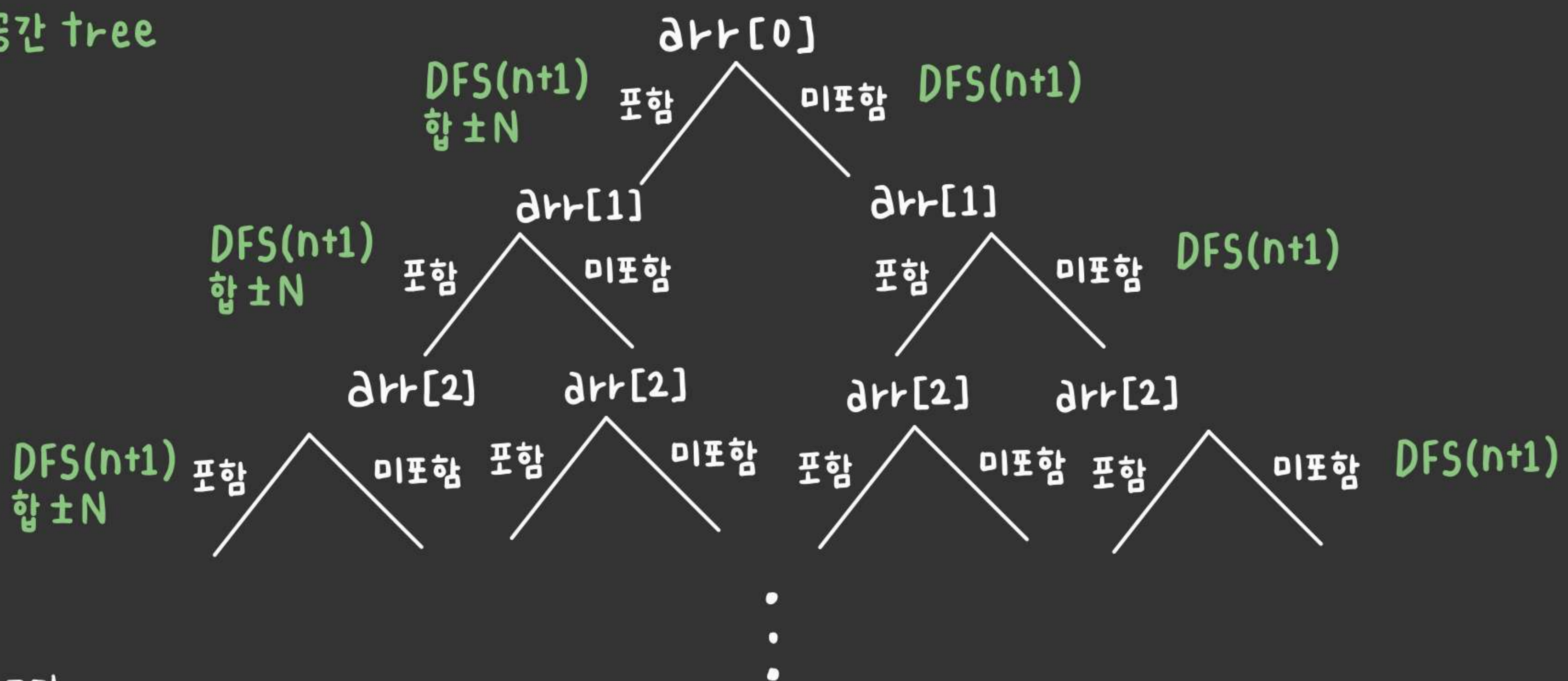
[입력]
 첫 번째 줄에 테스트 케이스의 수 T가 주어진다.
 각 테스트 케이스의 첫 번째 줄에는 2개의 자연수 N($1 \leq N \leq 20$)과 K($1 \leq K \leq 1000$)가 주어진다.
 두 번째 줄에는 N개의 자연수 수열 A가 주어진다. 수열의 원소인 N개의 자연수는 공백을 사이에 두고 주어지며, 1 이상 100 이하임이 보장된다.

[출력]
 각 테스트 케이스마다 '#x' (x는 테스트케이스 번호를 의미하며 1부터 시작한다)를 출력하고, 부분 수열의 합이 K가 되는 경우의 수를 출력한다.

예시) N=4 k=3
 arr = [1, 2, 1, 2]

arr의 부분집합 생성하여 합을 계산한다 (재귀)
 합이 K와 같으면 카운팅!

상태공간 tree



종료조건
 인덱스 = N

```
Cnt(경우의수)
def subset(i, ssum):
    if ssum > k: 가지치기
        return
    if i == N:
        if ssum == k:
            Cnt += 1
        return
    Subset(N+1, ssum + arr[i]) 포함한다
    Subset(N+1, ssum) 포함하지 않는다
```