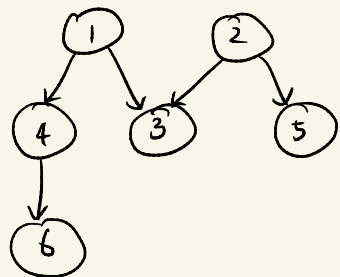


# 그래프 경로 (4871) - SWEA - DFS

V개 노드, E개의 간선  
특정한 두 개의 노드에 경로가 존재하는지 확인하기

⇒ DFS 사용



있으면 1  
없으면 0  
1 → 6 가는 경우 = 1  
노드는 1번부터 존재 한다.

- 입력
- ① Testcases
  - ② V 와 E 가 주어진다.
  - ③ 출발 → 도착을 알려준다.

- ④ 출발 S → 출발 G를 알려준다.
- 있으면 1  
없으면 0

V = 갯수 E = 줄 수

Matrix = [0 for i in range(V+1)]

for line in range(E):

i, j = tuple(map(int, input().split()))

Matrix[i][j] = 1

stack = [0 for i in range(V)]

top = -1

visited = [False for i in range(V+1)]

v = S

while True:

for j in range(len(matrix[v])):

if matrix[v][j] == 1 and visited[j] == False:

visited[j] = True

top += 1

stack[top] = matrix[v][j]

v = matrix[v][j]

break

if stack[0] == S and stack[-1] == G:

print(1) / n break

매트릭스 만들기

top = -1

stack과 visited를 default 값으로 초기화하고 S 인덱스를 True로 스택에 넣기 top += 1

순회하면서 stack에 살고 visited를 True로 바꾸기

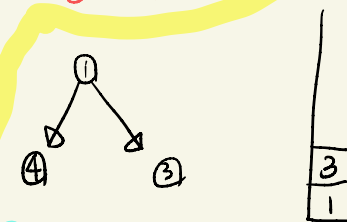
길이 막혀있으면 stack에서 값을 빼면서 돌아가기

만약 시작점에서 갈수 있는 곳이 없다면 (모든 곳이 True 라면) 0을 출력

혹은 top이 -1일 때

만약 스택의 마지막 인덱스가 G가 되는 순간 가는 것

내가 틀린 부분



이부분 그대로 하면 1로 돌아가는게 아니다.

따라서 3을 없애고 1로 가서 v에 값을 넣어준다

top을 이동하고

1) 길을 켜다

2) stack, visited를 최값으로 설정 'S'

3) for 문 돌린다.

① DFS를 켜다.

② stack이 S로 지나서 G로 끝나는 stack이 존재하는지 확인

있으면 1 없으면 0

행 열  
matrix[i][j] = 1

방향이 정해져 있다

하지만 지도 읽 (방향 없으면)

matrix[j][i] = 1을 해바람

①