BFS와 DFS

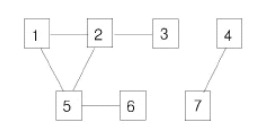
1. 바이러스( #2606 )

노드와 간선이 주어집니다.

아래 그림과 같이 점들이 연결된 관계가 주어졌을 때,

1번 노드가 바이러스에 감염되었다면, 연결된 점들은 모두 바이러스에 감염됩니다.

입력이 주어졌을 때, 1번 노드와 연결되어 감염된 노드들의 수를 출력하시오.



**입력**

**7**

**6**

**1 2**

**2 3**

**1 5**

**5 2**

**5 6**

**4 7**

**출력**

**4**

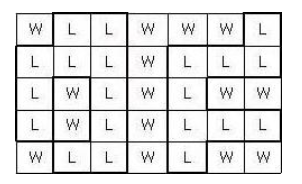
1. 보물섬( #2589 )

바다와 섬을 아래와 같이 스트링으로 표현했습니다.

바다라면 Water의 W를 섬이라면 Land의 L로 표시했습니다.

섬에서 한 점과 한 점의 위치가 가장 먼곳에 보물이 숨겨져 있습니다.

보물사이의 최단 거리를 구하세요.



**입력**

**5 7**

**WLLWWWL**

**LLLWLLL**

**LWLWLWW**

**LWLWLLL**

**WLLWLWW**

**출력**

**8**

**정답코드**

#2606

**from** **collections** **import** deque

computer = int(input())

connection = int(input())

graph = [[]**for** \_ **in** range (computer+1)]

visited = [0 **for** \_ **in** range (computer+1)]

**for** \_ **in** range (connection):

a,b = map(int,input().split())

graph[a].append(b)

graph[b].append(a)

q = deque()

q.append(1)

**while** q:

k = q.popleft()

visited[k] = 1

**for** i **in** graph[k]:

**if** visited[i] == 0:

q.append(i)

print(sum(visited)-1)

#2589

**from** **collections** **import** deque

n, m = map(int,input().split())

graph = [list(input().rstrip()) **for** \_ **in** range(n)]

maxi = 0

**for** y **in** range(n):

**for** x **in** range(m):

**if** graph[y][x] == *'L'*:

*# 방문 초기화*

visited = [[0 **for** \_ **in** range(m)] **for** \_ **in** range(n)]

dist = [[0 **for** \_ **in** range(m)] **for** \_ **in** range(n)]

*# BFS*

q = deque()

q.append((y,x))

visited[y][x] = 1

**while** q :

ey, ex = q.popleft()

*# 4 방향 탐색*

**for** dy, dx **in** [(0,1), (0,-1), (1,0), (-1,0)]:

ny, nx = ey + dy, ex + dx

**if** 0 <= ny < n **and** 0 <= nx < m **and** graph[ny][nx] == *'L'* **and** visited[ny][nx] == 0:

visited[ny][nx] = 1

dist[ny][nx] = max(dist[ey][ex] + 1, dist[ny][nx])

**if** maxi < dist[ny][nx]:

maxi = dist[ny][nx]

q.append((ny,nx))

print(maxi)

*# 이동은 상하좌우*

*# 육지만 이동 가능*

*# 육지 L 바다 W*

*# 보물은 가장 긴 시간이 걸리는 육지 두곳에 나뉘어 묻혀있다.*

*# 모든 L 에서 BFS로 거리를 계산하기*

*# 가장 거리가 먼곳 -> 정답*