7주차 주간학습보고서(5/3~5/9)

분류 1	전공		
팀 명	정융팡팡	제출자	안희승
학습 일시	2021.05.03 17:00 ~ 2021.05.03 19:00	마감 일시	2021.05.03 19:00
제출 일시	2021.05.07 10:18	수정 일시	2021.05.07 10:18

[2021-1 광운스터디그룹]

주간학습보고서

학습주제	DFS & BFS
학습장소	기념관 2층 로비
학습방법 및 학습개요	실전 감각을 키우기 위해 처음 한 시간은 전주에 풀었던 DFS & BFS 문제 하나와 처음 풀어보는 문제 총 두문제를 시간 내에 풀어보는 연습을 했습니다. DFS & BFS 문제(이전 주에 풀이) 1. 11724_연결요소의 개수 2. 7576_토마토 3. 2606_바이러스 4. 2629_양팔저울(공동문제) 시간 제한을 두고 풀었던 문제 1. 1012_유기농 배추 2. 2606_바이러스(다시 푼 문제)
다음 주 계획	**주제 : DFS&BFS(심화)** **일시 : 5.10** **장소 : 기념관 2층 로비** **보고서 제출자 : 김도연**

* 학습개요의 내용을 구체적으로 작성해 주세요.
이전 주에 풀었던 문제들을 리뷰하면서 DFS & BFS 문제는 대체로 노드로 연결된 그래프형태로 문제가 주어지거나 행렬 형태로 주어지는 문제가 많이 있는 것 같다는 점에 대해서 이야기하고 이 두가지 유형에 대한 준비를 충분히 해야 한다는 결론을 내림
보통 문제들에는 DFS & BFS 둘다 적용해서 문제를 풀이 할 수 있고 성능에 따른 차이도 그렇게 크지는 않지만 모든 경우의 수를 탐색해야하는 경우에는 DFS가 조금더 유리하고 그렇지 않은 경우에는 BFS를 사용하는 것이 더 좋다는 것을 구입한 교재 '이것이 코딩테스트이다'에서 확인할 수 있었음
〈그래프 형태의 문제〉
1. 연결요소의 개수

연결 요소의 개수 성공 분류

☆

실버 II

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
3 초	512 MB	43064	20555	13404	44.783%

문제

방향 없는 그래프가 주어졌을 때, 연결 요소 (Connected Component)의 개수를 구하는 프로그램을 작성하시오.

입력

첫째 줄에 정점의 개수 N과 간선의 개수 M이 주어진다. (1 ≤ N ≤ 1,000, 0 ≤ M ≤ N×(N-1)/2) 둘째 줄부터 M개의 줄에 간선의 양 끝점 u와 v가 주어진다. (1 ≤ u, v ≤ N, u ≠ v) 같은 간선은 한 번만 주어진다.

출력

첫째 줄에 연결 요소의 개수를 출력한다.

예제 입력 1 복사

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

예제 출력 1 복사

```
2
```

```python
import sys
input = sys.stdin.readline
N,M=map(int,input().split())

graph=[[] for _ in range(N+1)]
for _ in range(M):
    a,b=map(int,input().split())
    graph[a].append(b)
    graph[b].append(a)
visited=[0]*(N+1)

def bfs(v):
    q=[v]
    while q:
        v=q.pop(0)
        for i in graph[v]:
            if visited[i]==0:
                q.append(i)
                visited[i]=1

count=0
for i in range(1,N+1):
    if visited[i]==0:
        bfs(i)
        count+=1

print(count)
```

<최수지>

2. 바이러스

바이러스 성공 첫 분류

☆

실버 III

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	128 MB	62314	28813	19799	44.877%

문제

신종 바이러스인 웜 바이러스는 네트워크를 통해 전파된다. 한 컴퓨터가 웜 바이러스에 걸리면 그 컴퓨터와 네트워크 상에서 연결되어 있는 모든 컴퓨터는 웜 바이러스에 걸리게 된다.

예를 들어 7대의 컴퓨터가 <그림 1>과 같이 네트워크 상에서 연결되어 있다고 하자. 1번 컴퓨터가 웜 바이러스에 걸리면 웜 바이러스는 2번과 5번 컴퓨터를 거쳐 3번과 6번 컴퓨터까지 전파되어 2, 3, 5, 6 네 대의 컴퓨터는 웜 바이러스에 걸리게 된다. 하지만 4번과 7번 컴퓨터는 1번 컴퓨터와 네트워크상에서 연결되어 있지 않기 때문에 영향을 받지 않는다.

< 그림 1 >

어느 날 1번 컴퓨터가 웜 바이러스에 걸렸다. 컴퓨터의 수와 네트워크 상에서 서로 연결되어 있는 정보가 주어질 때, 1번 컴퓨터를 통해 웜 바이러스에 걸리게 되는 컴퓨터의 수를 출력하는 프로그램을 작성하시오.

입력

첫째 줄에는 컴퓨터의 수가 주어진다. 컴퓨터의 수는 100 이하이고 각 컴퓨터에는 1번 부터 차례대로 번호가 매겨진다. 둘째 줄에는 네트워크 상에서 직접 연결되어 있는 컴퓨터 쌍의 수가 주어진다. 이어서 그 수만큼 한 줄에 한 쌍씩 네트워크 상에서 직접 연결되어 있는 컴퓨터의 번호 쌍이 주어진다.

출력

1번 컴퓨터가 웜 바이러스에 걸렸을 때, 1번 컴퓨터를 통해 웜 바이러스에 걸리게 되는 컴퓨터의 수를 첫째 줄에 출력한다.

```python
# DFS
def dfs(graph, v, visited):
  visited[v] = True
  print(v, end=' ')
  for i in graph[v]:
    if not visited[i]:
      dfs(graph, i, visited)

# BFS
from collections import deque
def bfs(graph, start, visited):
  queue = deque([start])
  visited[start] = True
  while queue:
    v = queue.popleft()
    print(v, end=' ')
    for i in graph[v]:
      if not visited[i]:
        queue.append(i)
        visited[i] = True

n,m,v = map(int,input().split())
graph = [[] for _ in range(n+1)]
for _ in range(m):
  x,y = map(int, input().split())
  graph[x].append(y)
  graph[y].append(x)
for i in range(len(graph)):
  graph[i].sort()

visited = [False]*(n+1)
dfs(graph,v,visited)
print()
visited = [False]*(n+1)
bfs(graph,v,visited)
```

〈김도연〉
〈행렬 형태의 문제〉
1. 토마토

토마토

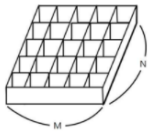
골드 | 효제 | 분큐

실버 I

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	256 MB	87320	30106	18844	33.386%

문제

철수의 토마토 농장에서는 토마토를 보관하는 큰 창고를 가지고 있다. 토마토는 아래의 그림과 같이 격자 모양 상자의 칸에 하나씩 넣어서 창고에 보관한다.



창고에 보관되는 토마토들 중에는 잘 익은 것도 있지만, 아직 익지 않은 토마토들도 있을 수 있다. 보관 후 하루가 지나면, 익은 토마토들의 인접한 곳에 있는 익지 않은 토마토들은 익은 토마토의 영향을 받아 익게 된다. 하나의 토마토의 인접한 곳은 왼쪽, 오른쪽, 앞, 뒤 네 방향에 있는 토마토를 의미한다. 대각선 방향에 있는 토마토들에게는 영향을 주지 못하며, 토마토가 혼자 저절로 익는 경우는 없다고 가정한다. 철수는 창고에 보관된 토마토들이 며칠이 지나면 다 익게 되는지, 그 최소 일수를 알고 싶어 한다.

토마토를 창고에 보관하는 격자모양의 상자들의 크기와 익은 토마토들과 익지 않은 토마토들의 정보가 주어졌을 때, 며칠이 지나면 토마토들이 모두 익는지, 그 최소 일수를 구하는 프로그램을 작성하라. 단, 상자의 일부 칸에는 토마토가 들어있지 않을 수도 있다.

입력

첫 줄에는 상자의 크기를 나타내는 두 정수 M,N이 주어진다. M은 상자의 가로 칸의 수, N은 상자의 세로 칸의 수를 나타낸다. 단, 2 ≤ M,N ≤ 1,000 이다. 둘째 줄부터는 하나의 상자에 저장된 토마토들의 정보가 주어진다. 즉, 둘째 줄부터 N개의 줄에는 상자에 담긴 토마토의 정보가 주어진다. 하나의 줄에는 상자 가로줄에 들어있는 토마토의 상태가 M개의 정수로 주어진다. 정수 1은 익은 토마토, 정수 0은 익지 않은 토마토, 정수 -1은 토마토가 들어있지 않은 칸을 나타낸다.

토마토가 하나 이상 있는 경우만 입력으로 주어진다.

출력

여러분은 토마토가 모두 익을 때까지의 최소 날짜를 출력해야 한다. 만약, 저장될 때부터 모든 토마토가 익어있는 상태이면 0을 출력해야 하고, 토마토가 모두 익지는 못하는 상황이면 -1을 출력해야 한다.

```python
import sys
from collections import deque

n, m = map(int, sys.stdin.readline().split())
li = []

for i in range(m):
    li.append(list(map(int, sys.stdin.readline().split())))

q = deque()

# 초기값을 큐에 넣어주기
for i in range(m):
    for j in range(n):
        if li[i][j] == 1:
            q.append((i, j))

count = 0

while q:
    count += 1
    for _ in range(len(q)):
        x, y = q.popleft()
        for a, b in (-1, 0), (1, 0), (0, -1), (0, 1):
            if 0 <= x+a < m and 0 <= y+b < n and li[x+a][y+b] == 0:
                li[x+a][y+b] = 1
                q.append((x+a, y+b))
            else:
                continue

breaker = False

# 안익은 토마토가 있다면 -1로 저장
for i in range(m):
    if breaker:
        break
    for j in range(n):
        if li[i][j] == 0:
            count = -1
            breaker = True

if count == -1:
    print(count)
else:
    print(count-1)
```

<안희승>

2. 유기농 배추

유기농 배추 성공 분류

☆

실버 II

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	512 MB	73288	27504	18590	35.821%

문제

차세대 영농인 한나는 강원도 고랭지에서 유기농 배추를 재배하기로 하였다. 농약을 쓰지 않고 배추를 재배하려면 배추를 해충으로부터 보호하는 것이 중요하기 때문에, 한나는 해충 방지에 효과적인 배추흰지렁이를 구입하기로 결심한다. 이 지렁이는 배추근처에 서식하며 해충을 잡아 먹음으로써 배추를 보호한다. 특히, 어떤 배추에 배추흰지렁이가 한 마리라도 살고 있으면 이 지렁이는 인접한 다른 배추로 이동할 수 있어, 그 배추들 역시 해충으로부터 보호받을 수 있다.

(한 배추의 상하좌우 네 방향에 다른 배추가 위치한 경우에 서로 인접해있다고 간주한다)

한나가 배추를 재배하는 땅은 고르지 못해서 배추를 군데군데 심어놓았다. 배추들이 모여있는 곳에는 배추흰지렁이가 한 마리만 있으면 되므로 서로 인접해있는 배추들이 몇 군데에 퍼져있는지 조사하면 총 몇 마리의 지렁이가 필요한지 알 수 있다.

예를 들어 배추밭이 아래와 같이 구성되어 있으면 최소 5마리의 배추흰지렁이가 필요하다.

(0은 배추가 심어져 있지 않은 땅이고, 1은 배추가 심어져 있는 땅을 나타낸다.)

1	**1**	0	0	0	0	0	0	0	0
0	**1**	0	0	0	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0	0
0	0	**1**	**1**	0	0	0	**1**	**1**	**1**
0	0	0	0	**1**	0	0	**1**	**1**	**1**

```python
import sys
from collections import deque

t = int(sys.stdin.readline())


for _ in range(t):
    m, n, k = map(int, sys.stdin.readline().split())
    li = [[0] * n for _ in range(m)]
    is_visited = [[False] * n for _ in range(m)]

    for i in range(k):
        a, b = map(int, sys.stdin.readline().split())
        li[a][b] = 1
    q = deque()
    count = 0

    for i in range(m):
        for j in range(n):
            if li[i][j] == 0 or is_visited[i][j]:
                continue
            else:
                q.append((i, j))
                count += 1
                is_visited[i][j] = True
                while q:
                    x, y = q.popleft()
                    for a, b in (1, 0), (-1, 0), (0, 1), (0, -1):
                        if 0 <= x + a < m and 0 <= y + b < n and li[x + a][y + b] == 1 and not is_visited[x+a][y+b]:
                            q.append((x + a, y + b))
                            is_visited[x+a][y+b] = True


    print(count)
```

* 학습활동 사진(필수)

온라인 활동시 : 시작 화면, 종료 화면(구성원 모두 나온 화면캡쳐)
오프라인 활동시 : 현장 활동 사진(구성원 및 활동 시간이 나오도록 촬영)