

A형 강의(2번째, DFS)

최신식

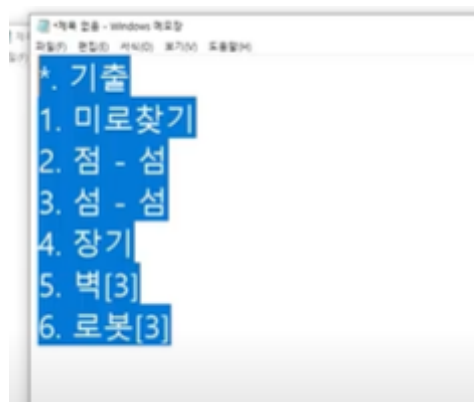
BFS 문제

하당정적.

- 종류별로 20개만 풀어보면....

20개

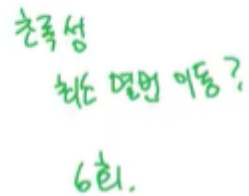
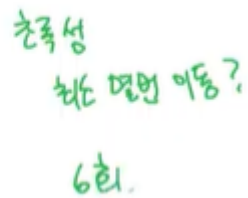
- 오늘 해야할 것



- 바이러스 문제, 토마토 문제 -> 블룸 문제(퍼지는 거라서)

- 블룸이 된다는 전제하에서....

map에 빨간색(사람)이 초록색으로 가려면 최소 몇번 이동해야 될까? 6번 (BFS함수로 구현해서 돌림)



- ## 능숙도 테스트 1번

문답지도 이미지 도구

①

벽

3

2 2 2

1

2

3 4 5 6

3점

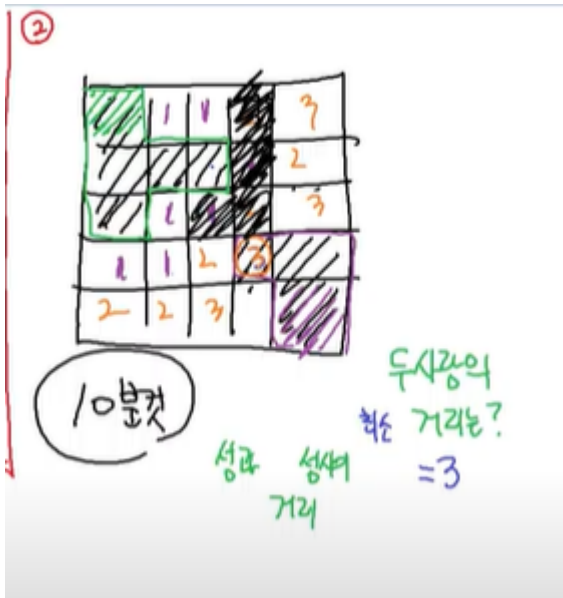
4점 - 0점

거리 = 7.

3점

능숙도 테스트 2번 : 천지창조 문제 (섬, 섬사이 거리는?)

0,0 n-1, n-1은 항상 고정되어 있고 섬의 모양은 달라짐



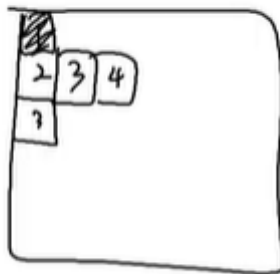
def bfs1()

def bfs2()

bfs를 두 번 돌려 구하기.

처음 bfs를 돌려서 섬 사이즈를 구하기.

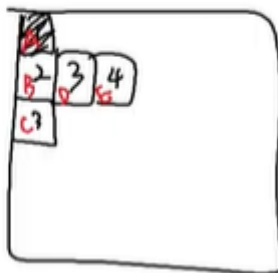
bfs 돌림. 섬 사이즈
구하는 bfs



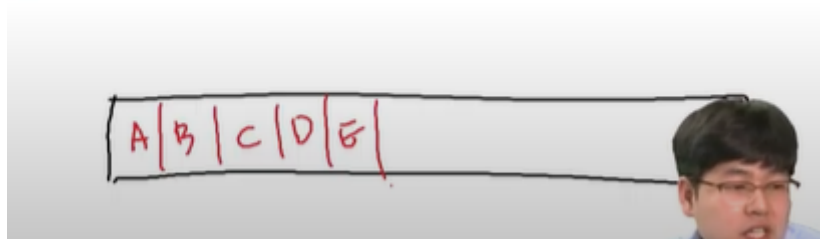
큐에 이렇게 쌓임

5 2회

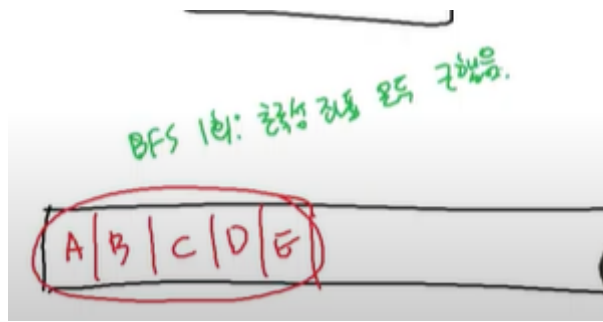
bfs 돌림. 섬 사이즈
구하는 bfs



1
?

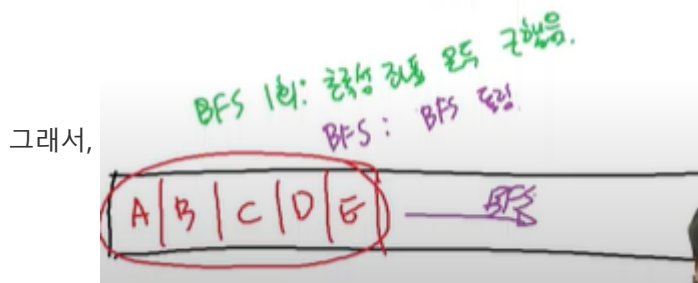
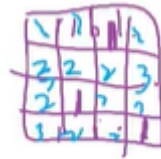


큐에 쌓인 채로 bfs를 끝냄



1회 다 돌리면 초록색 좌표를 다 구한 것임

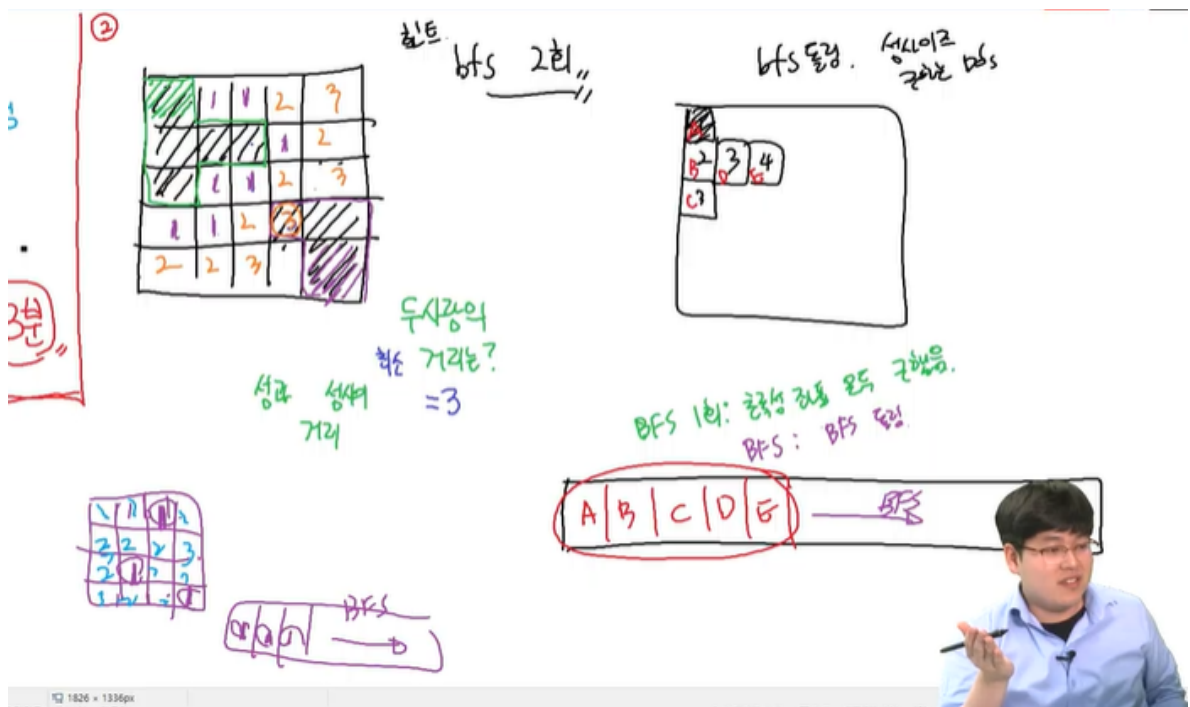
어제한 기본 문제 중에 씨앗이 여러개일때, 여러개의 좌표를 큐에 등록해 놓고 bfs를 돌리는 게 있었음.



그래서,

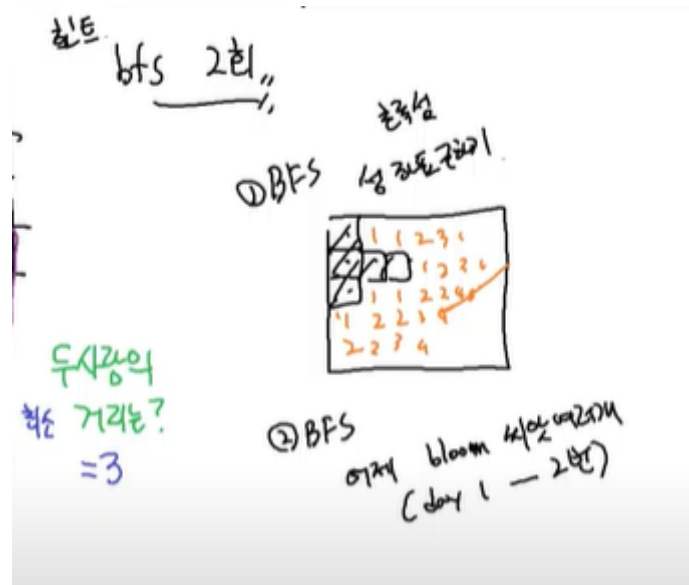
이 상태로 bfs를 돌리면 끝!

그럼

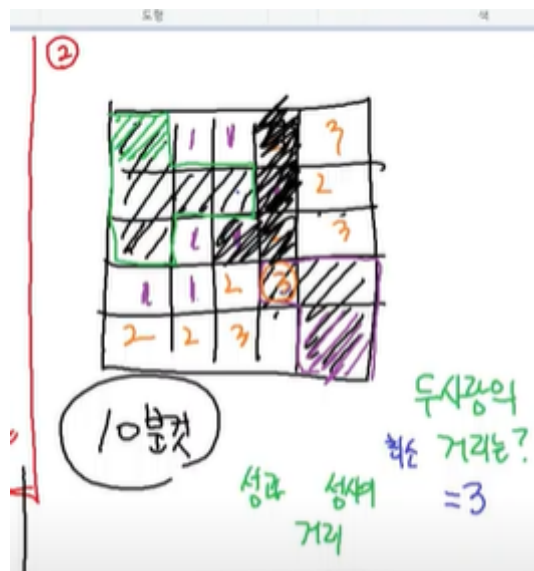


정리) 먼저 bfs로 초록색 좌표를 다 구한다. 최종 result에는 섬의 좌표가 다 들어 있어야 한다.

그 다음 bfs를 돌려서 구하면 됨



벽이 있는 경우도 있음



BFS 휴먼에러

- 1) 8방향 잘보기
- 2) 범위설정(같거나 크다 이런것들)
- 3) 변수 두번 넣지않았나 체크

두 사람의
최소 거리는?
=3

BFS
후면 예러
"1,2,3"

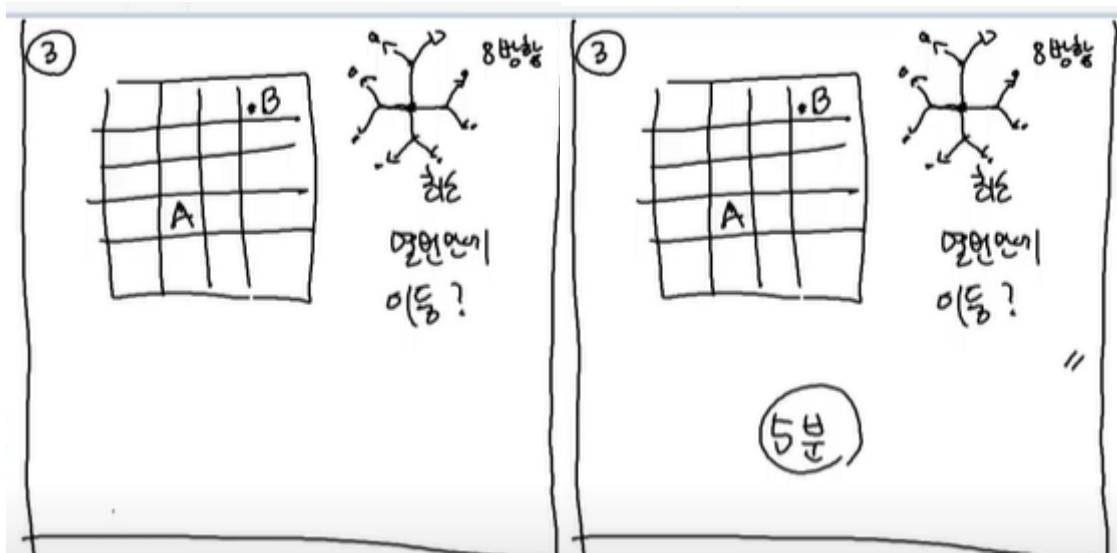
1 1 0 0 1 1 1
0 0 1 1 1 1 1

$0 \leq dx \leq N$ & $0 \leq dy \leq N$

---> i, j, m, n 변수이름 지양하기

ij1 | !!!!!
mnmnnm |



능숙도 테스트 3번 (A~B까지 몇번만에 갈 수 있나?)






A에서 B까지 몇번만에 갈 수 있나?

기출 14190 민코씨의 꽃밭

하루가 지나면 꽃이 핀다.

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

3일동안  생활만
die..

 말이 되지만,
주전)  양분을
다먹으면 die..

2	(1
((1
1	((

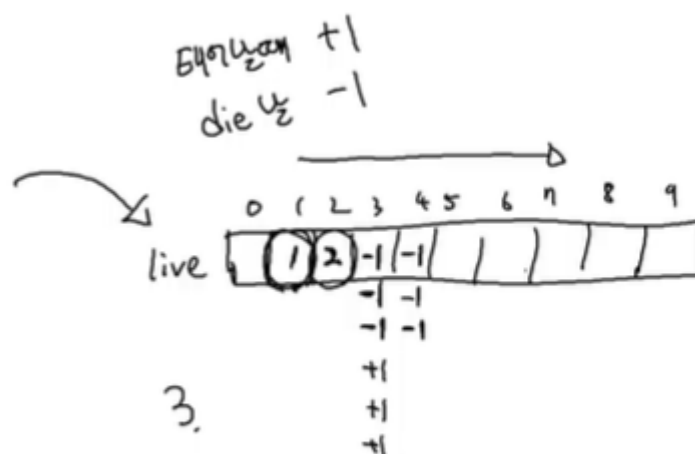
$$20\frac{1}{2} \quad 374.$$

가장 꽃이 많이 폈을 때가 언제인지, 꽃의 개수 출력

배열을 하나 만들어두고,

A hand-drawn 3x3 grid. The top row contains a circle with a dot, the number 2, a circle with a dot, a circle with a dot, and a circle with a dot. The middle row contains a circle with a dot, a circle with a dot, a circle with a dot, a circle with a dot, and a circle with a dot. The bottom row contains a circle with a dot, a circle with a dot, a circle with a dot, a circle with a dot, and a circle with a dot.

201 374.



첫째줄에는 테스트 케이스의 총 수가 입력됩니다.

각 테스트 케이스별로

첫째줄에는 밭의 세로 크기 N, 가로 크기 M 이 입력됩니다. ($1 \leq N, M \leq 500$)

둘째줄부터 N 줄에 걸쳐 각 줄마다 M 개의 정수가 공백으로 주어집니다.

해당 정수는 0 이상의 정수이며 해당 격자 땅의 비옥함을 나타냅니다. ($0 \leq \text{정수} \leq 1000,000$)

마지막 줄에는 민코씨가 처음 씨앗을 심게 되는 격자 지점이 행, 열 형태로 입력됩니다. ($0 \leq \text{행} < N, 0 \leq \text{열} < M$)

250000+1000000 하면 live배열의 최대 크기?

```
1  from collections import deque
2
3  tc = int(input())
4  ans = []
5  for TC in range(1, tc + 1):
6      N, M = map(int, input().split())
7      MAP = [list(map(int, input().split())) for _ in range(N)]
8      R, C = map(int, input().split())
9      visited = [[0 for _ in range(N)] for _ in range(M)]
10
11     qu = deque()
12     qu.append((R, C, 1))
13     visited[R][C] = 1
14
15     dr = [-1, 1, 0, 0]
16     dc = [0, 0, -1, 1]
17     liveCnt = [0 for _ in range(1250001)]
18     while qu:
19         nowR, nowC, nowLiveDay = qu.popleft()
20
21         liveCnt[nowLiveDay] += 1
22         liveCnt[nowLiveDay + MAP[nowR][nowC]] -= 1
23
24         for t in range(4):
25             nr = nowR + dr[t]
26             nc = nowC + dc[t]
27             if nr < 0 or nc < 0 or nr >= N or nc >= M: continue
28             if visited[nr][nc] == 1: continue
29             if MAP[nr][nc] == 0: continue;
30             visited[nr][nc] = 1
31             qu.append((nr, nc, nowLiveDay + 1))
32
33     now_cnt = 0
34     max_cnt = -1
35     maxDay = -1
36     for day in range(1, 1250001):
37         now_cnt += liveCnt[day]
38         if max_cnt < now_cnt:
39             max_cnt = now_cnt
40             maxDay = day
41     ans.append('#{0} {1}일 {2}개 '.format(TC, maxDay, max_cnt))
42
43 for answer in ans:
44     print(answer)
```

- 블룸이랑 같음(노란색박스 제거하면)


```

1 from collections import deque
2
3 tc = int(input())
4 ans = []
5 for TC in range(1, tc + 1):
6     N, M = map(int, input().split())
7     MAP = [list(map(int, input().split())) for _ in range(N)]
8     R, C = map(int, input().split())
9     visited = [[0 for _ in range(N)] for _ in range(M)]
10
11     qu = deque()
12     qu.append((R, C, 1))
13     visited[R][C] = 1
14
15     dr = [-1, 1, 0, 0]
16     dc = [0, 0, -1, 1]
17     liveCnt = [0 for _ in range(1250001)]
18     while qu:
19         nowR, nowC, nowLiveDay = qu.popleft()
20         liveCnt[nowLiveDay] += 1
21         liveCnt[nowLiveDay + MAP[nowR][nowC]] -= 1
22
23         for t in range(4):
24             nr = nowR + dr[t]
25             nc = nowC + dc[t]
26             if nr < 0 or nc < 0 or nr >= N or nc >= M: continue
27             if visited[nr][nc] == 1: continue
28             if MAP[nr][nc] == 0: continue
29             visited[nr][nc] = 1
30             qu.append((nr, nc, nowLiveDay + 1))
31
32     now_cnt = 0
33     max_cnt = -1
34     maxDay = -1
35     for day in range(1, 1250001):
36         now_cnt += liveCnt[day]
37         if max_cnt < now_cnt:
38             max_cnt = now_cnt
39             maxDay = day
40     ans.append('#{0} {1}일 {2}개'.format(TC, maxDay, max_cnt))
41
42 for answer in ans:
43     print(answer)
44

```

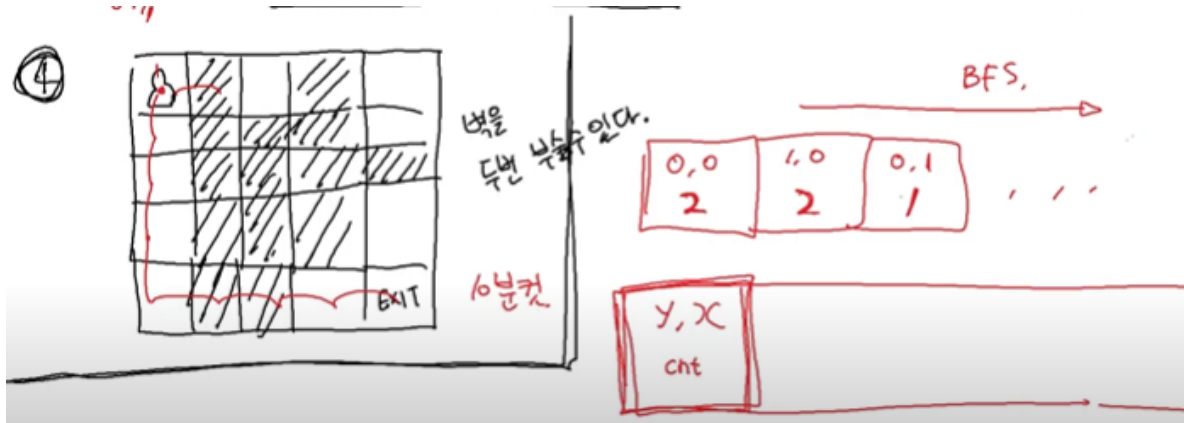
BFS.

능숙도 테스트 4번 (미로찾기의 변형 문제)

사람이 있고, 벽을 최대 두번부실 수 있다. 최소 몇번 이동해서 exit에 도착할 수 있는지?

큐에서 한 노드에 무슨 값이 들어갈 지를 설계해야한다.

좌표와 cnt(벽부시기횟수)



used 배열을쓰는이유가 한 번 온 곳을 다신 못오게 하는 게 목표

하지만 여기는 used[y][x][cnt] 세개를 운영해줘야함...



능숙도 테스트 5번 로봇 (10분컷)

3차원 `used`...ㅎ.ㅎ.. 좌표y, x, 방향(bang)

노드에는 좌표, 방향, 커멘드 횟수

`used[y][x][bang]`

y, x
bang, lev

앞으로 간다 / 왼쪽 / 오른쪽 세개 선택 가능

도착하는데 최소 9번이 필요함

