

[BOJ]1012 유기농 배추

0327

예시) M, N, K = 5, 3, 6

arr =

0	2
1	2
2	2
3	2
4	2
4	0

i) 배추를 이차원 배열에 배치한다 (배추가 있으면, 1)

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

ii) 방문리스트

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

iii) delta를 이용해 탐색
— 방문한 적이 없고
— 배추가 존재한다

[BOJ]2583 영역구하기

예시) M, N, K = 5, 7, 3

Paper =

0	5
1	4
2	3
3	2
4	1
5	0

1	2	3	4	5	6	7	
0	1	2	3	4	5	6	

원·아래 오·위

arr =

0	2	4	4
1	1	2	5
4	0	6	2
j	i	j	i

$M - arr[i][3] \leq row < M - arr[i][1]$
(i)
 $arr[i][0] \leq column < arr[i][2]$
(j)

↪ i) 이 영역만큼 직사각형이 있다 → 배열에 표시하기
ii) 배열 순회하면서 분리된 영역 탐색!

[BOJ]10026 적록색약

예시) N = 5

arr =

R	R	R	B	B
G	G	B	B	B
B	B	B	R	R
B	B	R	R	R
R	R	R	R	R

i) 방문배열 생성

(0,0)부터 탐색시작

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

ii) 탐색!

종료조건: 모든 칸 탐색

적록색약 X : 상·하·좌·우에 같은 글자가 있으면 이동!
(범위 & 방문체크)

적록색약 : 현재 점점의 값이 R이나 G일 경우 상·하·좌·우에 R이나 G면 이동!
현재 점점의 값이 B일 경우 상·하·좌·우에 B면 이동!
(범위 & 방문체크)

[BOJ] 2589 보물섬

예시) $N, M = 5, 7$

아바 =

W	L	L	W	W	W	L
L	L	L	W	L	L	L
L	W	L	W	L	W	W
L	W	L	W	L	L	L
W	L	L	W	L	W	W

보물은 최단거리가 가장 긴 곳에 위치한다

∴ 정점간 거리가 가장 먼 두 정점 찾기 → 가장 깊은 깊이를 찾자!
(깊이 우선 탐색)

```

max_depth = 0
def dfs(i, j, d):
    if max_depth < d:
        max_depth = d
    visit[i][j] = 1
    for di, dj in delta:
        if 범위, 'L', 방문여부 확인
            ni = i + di
            nj = j + dj
            dfs(ni, nj, d + 1)

```

→ 그러나.. 돌아서 가는 것을 막을 수 없다 

bfs로 수정

i) 모든 육지정점을 탐색의 시작점으로 설정

ii) delta, 육지, 방문여부 확인하여 인접육지 탐색

[BOJ] 2606 바이러스

예시) $N=7$ (정점수) $M=6$ (간선수)

아바 =

1	2
2	3
1	5
5	2
5	6
4	7



i) 인접배열

	2,5	1,3,5	2	7	2,6	5	4
--	-----	-------	---	---	-----	---	---

ii) 방문리스트

0	1						
---	---	--	--	--	--	--	--

iii) 완전탐색

→ 방문하지 않고 인접한 곳

→ 방문한 정점수를 방문리스트에 기록

[BOJ] 7576 토마토

예시) $M, N = 6, 4$

$$arr = \begin{array}{|c|c|c|c|c|c|} \hline 1 & -1 & 0 & 0 & 0 & 0 \\ \hline 0 & -1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & -1 & 0 \\ \hline 0 & 0 & 0 & 0 & -1 & 1 \\ \hline \end{array}$$

i) 탐색의 시작점 찾기 : 값이 1인 좌표

 $(0, 0), (3, 5) \rightarrow$ 초기세팅

(방에추가, 방문배열 표시)

ii) 방문배열

$$\begin{array}{|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 \\ \hline \end{array}$$

직접코드 구현해보니 필요 X
 arr로 방문여부 확인 가능!

iii) 탐색 시작 : delta 이동(범위체크) & 토마토가 있고 방문하지 않은 곳

iv) 탐색 후, arr에 0이 있으면 -1을 출력

탐색 후, arr에 0이 없으면 마지막으로 토마토를 탐색한 날을 출력

탐색 = 익는다

[BOJ] 7569 토마토

7576문제에서 앞,뒤가 추가되었다. 총마다 2차원 배열을 가진다

예시) $M, N, H = 5, 3, 2$

$$arr = \begin{array}{|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 \\ \hline 2 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 & 0 \\ \hline 2 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

i) delta

상: $(0, -1, 0)$ 앞: $(-1, 0, 0)$ 하: $(0, 1, 0)$ 뒤: $(1, 0, 0)$ 좌: $(0, 0, -1)$ 우: $(0, 0, 1)$ $arr[level][row][column]$ ex) $arr[1][1][2] = 0$

ii) delta 범위체크

상·하·좌·우인 경우

 $0 \leq row < N$ $0 \leq column < M$

앞·뒤인 경우

 $0 \leq level < h$

나머지는 7576번 문제와 동일하다