

**백준 2468 안전영역**

202020988 조아영

# 001 문제 소개

# 002 아이디어

# 003 코드 설명

# # 문제 소개

어떤 지역의 높이 정보를 파악한 후, 그 지역에 많은 비가 내렸을 때 물에 잠기지 않는 안전한 영역이 최대 몇개가 만들어지는 지를 조사 하려고 한다.

내리는 비의 양에 따라 일정 높이 이하의 모든 지점은 물에 잠긴다고 가정

6	8	2	6	2
3	2	3	4	6
6	7	3	3	2
7	2	5	3	6
8	9	5	2	7

6	8	2	6	2
3	2	3	4	6
6	7	3	3	2
7	2	5	3	6
8	9	5	2	7

6	8	2	6	2
3	2	3	4	6
6	7	3	3	2
7	2	5	3	6
8	9	5	2	7

행, 열의 크기가 N=5인  
지역의 높이 정보

높이가 4 이하인 지점은 물에 잠김  
안전 영역 (5개)

안전 영역 (4개)

비의 양에 따라 물에 잠기지 않는 안전한 영역의 개수는 다름

N의 값과 지역의 높이 정보가 주어졌을 때, 물에 잠기지 않는 안전한 영역의 최대 개수를 계산하여 출력

## # 아이디어

높이를 0부터 ~ 각 지역의 최대 높이까지 반복  
( 2차원 배열에서의 최댓값 구해야 함 )

각 높이 마다 잠기는 지역, 잠기지 않는 지역 체크  
( 잠김 유무 체크할 2차원 배열 생성 )

잠기지 않는 영역의 개수를 BFS로 셈

# # 코드 설명

```
from collections import deque
import sys
```

```
dx = [1, -1, 0, 0]   인접 영역 ( 앞, 뒤, 오, 왼 ) 확인 하기 위해
dy = [0, 0, -1, 1]
```

```
def bfs(i, j):
    queue = deque()
    check[i][j] == True   방문 표시
    queue.append([i, j])  큐에 삽입
```

```
while queue:
    x, y = queue.popleft()  큐에서 하나 꺼내, x, y 위치 저장
    for k in range(4):
        nx = x + dx[k]
        ny = y + dy[k]      영역 범위 안이고 방문하지 않은 경우
        if 0 <= nx < N and 0 <= ny < N and check[nx][ny] == False:
            check[nx][ny] = True   방문 표시
            queue.append([nx, ny])  큐에 삽입
```

```

if __name__ == "__main__":
    N = int(sys.stdin.readline())                지역의 높이 입력 ( 2차원배열 )
    matrix = [list(map(int, sys.stdin.readline().split())) for _ in range(N)]
    height = max(map(max, matrix))              지역 높이의 최댓값 저장
    num = []

    for i in range(0, height+1):                높이 0 ~ 최대높이값
        check = [[False for _ in range(N)] for _ in range(N)]    잠김 유무 체크
        for j in range(N):                      전체 영역을 돌면서
            for k in range(N):
                if matrix[j][k] <= i:            지역의 높이가 i이하인 경우
                    check[j][k] = True          잠김(방문) 표시

    cnt = 0
    for j in range (N):                        전체 영역을 돌면서
        for k in range (N):
            if check[j][k] == False:            잠기지(방문하지) 않은 곳인 경우
                bfs(j, k)                      bfs 호출
                cnt += 1                        영역의 개수

    num.append(cnt)
    print(max(num))                            영역의 개수의 최댓값 출력

```

감사합니다