

백준 2583 영역 구하기

202020988 조아영

# 001 문제 소개

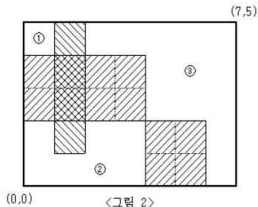
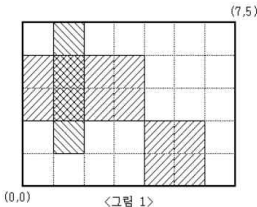
# 002 아이디어

# 003 코드 설명

# # 문제 소개

눈금의 간격이 1인  $M \times N$  ( $M, N \leq 100$ ) 크기의 모눈종이가 있다. 이 모눈종이 위에 눈금에 맞추어  $K$ 개의 직사각형을 그릴 때, 이들  $K$ 개의 직사각형의 내부를 제외한 나머지 부분이 몇 개의 분리된 영역으로 나누어진다.

예를 들어  $M=5$ ,  $N=7$  인 모눈종이 위에 <그림 1>과 같이 직사각형 3개를 그렸다면, 그 나머지 영역은 <그림 2>와 같이 3개의 분리된 영역으로 나누어지게 된다.



<그림 2>와 같이 분리된 세 영역의 넓이는 각각 1, 7, 13이 된다.

출력

**입력**  $M$ ,  $N$ 과  $K$  그리고  $K$ 개의 직사각형의 좌표가 주어질 때,  $K$ 개의 직사각형 내부를 제외한 나머지 부분이 몇 개의 분리된 영역으로 나누어지는지, 그리고 분리된 각 영역의 넓이가 얼마인지를 구하여 이를 출력하는 프로그램을 작성하시오.

## # 아이디어

visited 2차원 배열 사용  
=> 입력된 직사각형 내부를 모두 방문 표시 !

영역 전체를 반복하여  
방문 하지 않은 부분에서 bfs 함수 호출

=> 영역의 개수, 넓이 한번에 구할 수 있음

# # 코드 설명

```
if __name__ == "__main__":  
    M, N, K = map(int, sys.stdin.readline().split())  
    visited = [[False] * N for _ in range(M)]  
  
    for i in range(K): K개의 직사각형 입력  
        space = list(map(int, sys.stdin.readline().split()))  
        for j in range(space[0], space[2]):  
            for k in range(space[1], space[3]):  
                visited[k][j] = True 방문 표시  
  
    result = []  
    for i in range(M): 전체 영역을 돌면서  
        for j in range(N):  
            if visited[i][j] == False: 방문하지 않은 곳인 경우  
                result.append(bfs(i,j)) bfs 함수 호출  
                result엔 영역의 넓이 저장  
  
    print(len(result)) 영역의 개수  
    result.sort()  
    for i in result:  
        print(i, end=" ") 영역의 넓이(오름차순)
```

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

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

```
def bfs(i, j):
    visited[i][j] = True   방문 표시
    queue = deque()
    queue.append([i,j])   큐에 삽입
    cnt = 1   영역의 넓이를 셀 변수 초기화

    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 < M and 0 <= ny < N and visited[nx][ny] == False:
                visited[nx][ny] = True   방문 표시
                queue.append([nx, ny])   큐에 삽입
                cnt += 1   영역의 넓이 1 증가

    return cnt
```

감사합니다