



분할 정복

분할정복법 (Divide and Conquer)

: 분할정복법은 주어진 문제를 작은 사례로 나누고, 각각의 작은 문제들을 해결하여 정복하는 방법이다.

분할정복법은 문제의 사례를 2개 이상의 더 작은 사례로 나눈다. 이 때, 나눈 작은 사례에서 **해답을 바로 얻을 수 있을 때까지** 나눈다.

분할 정복법은 하향식 접근 방법으로 최상위 사례의 답을 아래로 내려가면서 작은 사례에 대한 답을 구함으로서 구한다. 또한, 동적 계획법과는 달리 가장 작은 단위의 문제로 만든 다음 해결하고 병합하는 과정에서 sub problem의 중복이 없다.

예시로 백준의 색종이 만들기 문제를 풀어보자.

문제

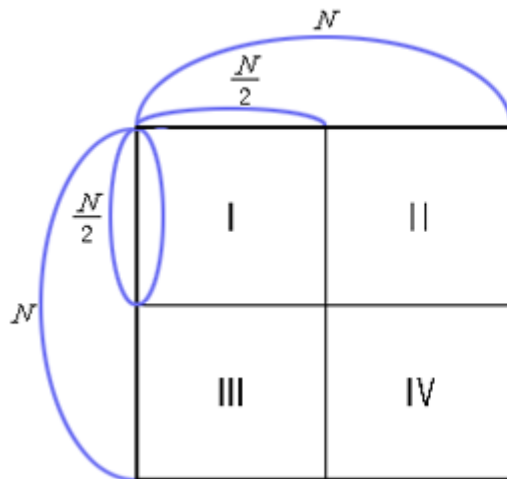
전체 $N \times N$ 의 크기를 갖는 종이를 일정한 규칙에 따라 잘라, 다양한 크기를 가진 정사각형 모양의 하얀색(0) 또는 파란색(1) 색종이를 만든다.

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

<그림 1> 8×8 종이

이 때, 나눈 부분의 전체 종이가 모두 같은 색으로 칠해져 있지 않으면, 가로와 세로 중간 부분을 잘라서 똑같은 크기의 네 개의 색종이로 나눈다. 예를 들어, $N \times N$ 크기의 색종이가 모두 같은 색이 아니라면, $N/2 \times N/2$ 크기로 나눈다.

이와 같은 과정을 잘라진 종이가 모두 하얀색 또는 모두 파란색으로 칠해져 있거나, 하나의 정사각형 칸이 되어 더 이상 자를 수 없을 때까지 반복한다.



<그림 2>

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

<그림 3> 처음 나눈 후의 상태

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

<그림 4> 두 번째 나눈 후의 상태

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

<그림 5> 최종적으로 나누어진 색종이들

입력으로 주어진 종이의 한 변의 길이 N 과 각 정사각형칸의 색(하얀색 또는 파란색)이 주어질 때 잘라진 하얀색 색종이와 파란색 색종이의 개수를 구하는 문제이다.

풀이

위의 <그림 2>와 같이, 전체 $N \times N$ 크기의 색종이를 $N/2 \times N/2$ 로 점점 잘라가며, 결국에는 1×1 크기가 될 때까지 <조건1>를 반복한다. 하지만 중간에 자른 색종이의 색이 모두 같다면 <조건2> 그 부분은 해당 색에 +1을 해주고, 종료한다.

따라서 `divide_and_conquer` 이라는 함수를 재귀적으로 활용하여 전체 큰 정사각형을 4개로 나누어가며 위 두 조건에 해당할 때까지 재귀를 반복한다.

소스코드

```
import sys

def divide_and_conquer(x, y, n):
    # x: start_x, y: start_y, n: 자를 크기
    global count
    isSame = True # 모든 색이 같은지 확인하는 변수
    color = papers[x][y] #처음 색을 color에 저장하고, 비교함.

    for i in range(x, x+n):
        if not isSame:
            break
    for j in range(y, y+n):
        if color != papers[i][j]:
            isSame = False
            divide_and_conquer(x, y, n//2) # 2사분면
            divide_and_conquer(x, y + n//2, n//2) #3사분면
            divide_and_conquer(x + n//2, y, n//2) #1사분면
            divide_and_conquer(x+ n//2, y+ n//2, n//2) #4사분면
            break

    if isSame:
        count[color] += 1

N = int(sys.stdin.readline().strip())
papers = []
for _ in range(N):
    papers.append(list(map(int,sys.stdin.readline().split())))
count = {0:0, 1:0}
divide_and_conquer(0,0,N)
print(count[0])
print(count[1])
```