Divide & Conquer

[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]

8x8 Count=0 n= int(input())

```
for i in range(n):
    Origin_Matrix.append(list(input().split(" ")))
```

```
fun check_색종이(x, y, n):
    Check = Origin_Matrix[x][y]
    for i in range(x, x+n)
        for j in range(y, y+n)
    If Check와 다르게 하나라도
    나오면:
```

⟹ 다음장

index 범위의 이유: x, y가 항상 시작값이기 때문에 끝값 전체를 검사하려고 +n, +n/2, +n/3, ... 을 해주어야 한다.

색종이 만들기

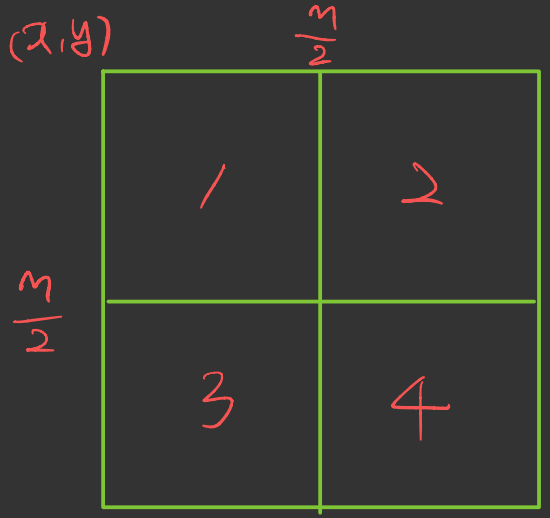
check_색종이(x, y, n//2): 1사분면

" (x, y+n//2, n//2) 2사분면

" (x+n//2, y, n//2) 3사분면

" (x+n//2, y+n//2, n//2) 4사분면

⇒ 재귀 돌리다 최소 단위로 devide



종이의 개수

```
def check_color_paper(x,y,n):
    check = origin_matrix[x][y]
    for i in range(x,x+n):
        for j in range(y,y+n):
            if check != origin_matrix[i][j]:
                check_color_paper(x,y,n//3)          # 1사분면
                check_color_paper(x,y+n//3,n//3)     #2사분면
                "          (x,y+2*n//3,n//3)         #3사분면
                "          (x+n//3,y,n//3)           #4사분면
                :                                    :
```

```
def check_color_Paper (cont.)

    if check == -1 :
        minus_count += 1
    elif check == 0 :
        zero_Count += 1
    elif check == 1 :
        Plus_Count += 1
```

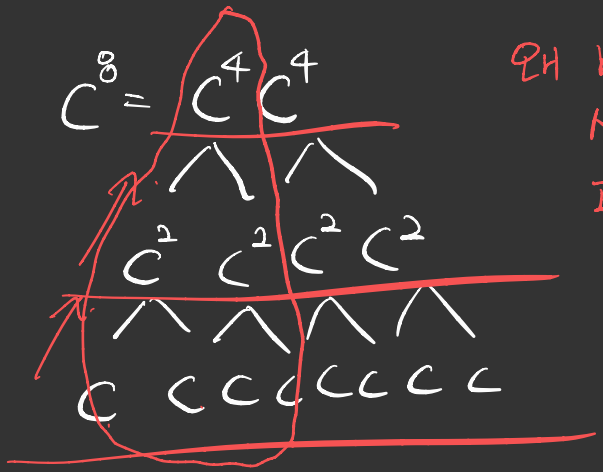
곱셈

* 분할정복을 이용한 거듭제곱 *

C^n 연산은 C를 n번 곱하므로 $O(n)$이다.

그러나 분할정복을 사용하면 $O(\log n)$으로 시간복잡도를 줄일 수 있다.

$$C^n = \begin{cases} C^{\frac{n}{2}} C^{\frac{n}{2}} & (n\text{은 짝수}) \\ C^{\frac{n-1}{2}} C^{\frac{n-1}{2}} C & (n\text{은 홀수}) \end{cases}$$



왜 return하는 과정은
시간복잡도에
고려를 안 할까?

곱셈 cont.

왜 각 연산마다 "%c"를 해야하지?

$$(A+B)\%M = ((A\%M)+(B\%M))\%M$$

$$(A*B)\%M = ((A\%M)*(B\%M))\%M$$

$$(A-B)\%M = ((A\%M)-(B\%M)+M)\%M$$

이항 계수3

* 페르마의 소정리 *

$a^p \% p = a \% p$ (p는 소수, a는 정수)

a와 p가 서로소이면

$a^{p-1} \% p = 1 \ (\% p)$

$a \cdot a^{p-2} \equiv 1 \pmod p$

a^{p-2} : a의 모듈로 연산에 대한 역원

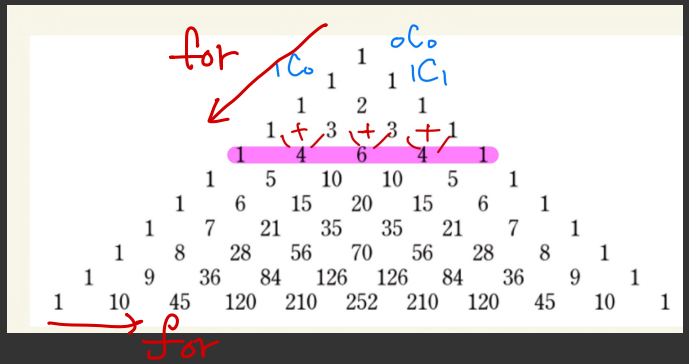
$${}_nC_k = \frac{n!}{k!(n-k)!}$$

$\frac{1}{k!(n-k)!}$ 을 어떻게 구하지?

왜 파스칼의 삼각형을 이용하여 풀면 안 될까?

이 문제에서 Input N의 범위가 크기 때문.

시간복잡도: $O(n^2)$ (∵ 이중 for문)



$[k!(n-k)!]^{p-2}$ ⇒ $k!(n-k)!$ mod 연산의 역원

$\frac{n!}{k!(n-k)!} \% p$

$O(n)$

$\frac{n!}{k!(n-k)!} \cdot k!(n-k)! \cdot [k!(n-k)!]^{p-2} \% p$

$O(\log n)$

이항 계수3 Cont.

" $nC_K \,\%\, P = \left(\frac{n!}{k!(n-k)!}\right) \%\, P = \frac{n! \,\%\, P}{(k!(n-k)!) \%\, P}$ " ← 이렇게 했을 경우 문제점

나머지가 0이되면
error가 발생한다.