

# 백준 2655, 가장 높은 탑 쌓기

Step 1

입력 (S: 빙면 높이, H: 높이, W: 너비,  
pos: 정렬 순 차례)

	0	1	2	3	4
S	25	4	9	16	1
H	3	4	2	2	5
W	4	6	3	5	2
pos	0	1	2	3	4

입력 높이를  
내림차순으로 정렬

정렬 후 (앞으로 블록을 쌓을 때, 높이만 바꾸면 됨)

	0	1	2	3	4
S	25	16	9	4	1
H	3	2	2	4	5
W	4	5	3	6	2
pos	0	3	2	1	4

dp

maxH : 해당 블록을 제일 첫째 놓을 때, 힙에 최대 높이  
prev : 해당 블록을 놓았을 때의 블록의 index  
pos : 해당 블록의 현재 위치

	0	1	2	3	4
maxH	0	0	0	0	0
prev	-1	-1	-1	-1	-1
pos	0	0	0	0	0

Step 2

	0	1	2	3	4
S	25	16	9	4	1
H	3	2	2	4	5
W	4	5	3	6	2
pos	0	3	2	1	4

Step 3

	0	1	2	3	4
S	25	16	9	4	1
H	3	2	2	4	5
W	4	5	3	6	2
pos	0	3	2	1	4

Step 4

	0	1	2	3	4
S	25	16	9	4	1
H	3	2	2	4	5
W	4	5	3	6	2
pos	0	3	2	1	4

	0	1	2	3	4
maxH	3	0	0	0	0
prev	-1	-1	-1	-1	-1
pos	0	0	0	0	0

0번 블록은 끝에 놓을 것으 둘 4 번으로  
maxH에 3, pos에 0, prev에 -1 대입

	0	1	2	3	4
maxH	3	2	0	0	0
prev	-1	-1	-1	-1	-1
pos	0	3	0	0	0

X  
1번 블록  
0번 블록

1번 블록은 0번보다 높아서 4개므로  
0번을 끝에 둘 수 없어서 바로 대입,  
현재 위치 3로 pos에 대입

	0	1	2	3	4
maxH	3	2	5	0	0
prev	-1	-1	0	-1	-1
pos	0	3	2	0	0

O  
2번 블록  
0번 블록  
1번 블록

2번 블록은 0과 1 블록 끝에 둘 수 있고  
dp[2], maxH에 pos + dp[2], maxH에 대입  
pos에 현재 위치 2를 대입  
prev에는 바로 끝의 블록의 index인 0을 대입

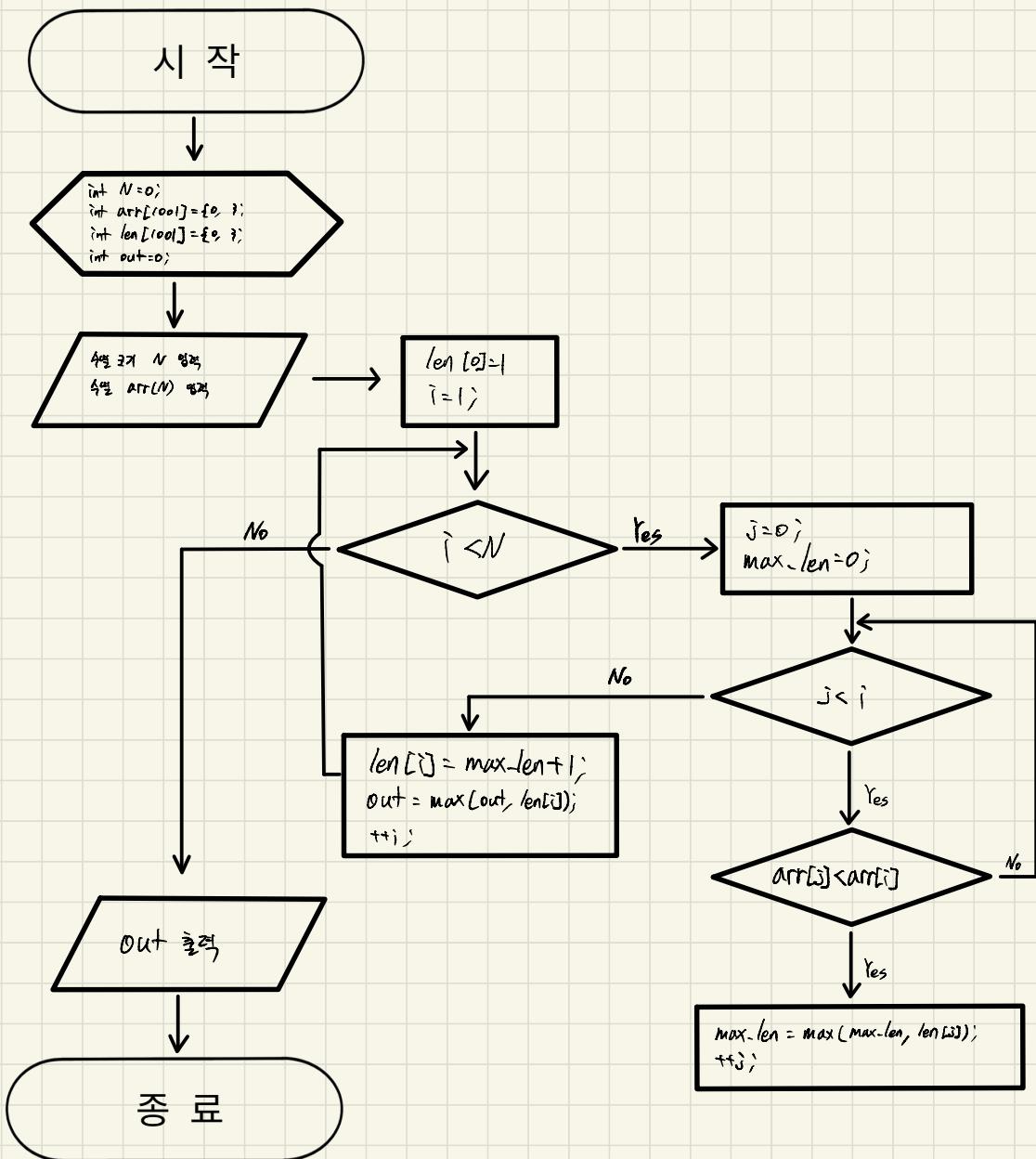
Step 5 4단계 수행 결과

	0	1	2	3	4
maxH	3	2	5	4	10
prev	-1	-1	0	-1	-1
pos	0	3	2	1	4

Step 6 출력

```
pseudo code
while (true)
{
    printf("%d\n", cur.dp.pos);
    if (cur.dp.prev == -1)
    {
        break;
    }
    cur.dp = dp[cur.dp.prev];
}
```

# 백준 11053, 가장 긴 증가하는 수열



# 백준 1904, 01타일

$$a_1 = 1 \quad \boxed{1}$$

$$a_2 = 2 \quad \boxed{1} \quad \boxed{1} \quad \boxed{0} \quad \boxed{0}$$

$$a_3 = 3 \quad \boxed{\phantom{0}} \quad \boxed{1} \quad + \quad \boxed{\phantom{0}} \quad \boxed{0} \quad \boxed{0}$$

$\downarrow \quad \quad \quad \downarrow$   
 $a_1 \quad + \quad a_2$

$$a_4 = 5 \quad \boxed{\phantom{0}} \quad \boxed{1} \quad + \quad \boxed{\phantom{0}} \quad \boxed{0} \quad \boxed{0}$$

$\downarrow \quad \quad \quad \downarrow$   
 $a_3 \quad + \quad a_2$

$$\text{정의: } a_n = a_{n-1} + a_{n-2}$$

## 백준 16395, 파스칼의 삼각형

$$\text{정의: } nCr = n-rCr + n-rCr-1$$

pseudo code

$$arr[0][0] = 1;$$

$$arr[1][0] = 1;$$

$$arr[2][0] = 1;$$

for (int i=2; i < N; ++i)

{

    for (int j=0; j < N; ++j)

{

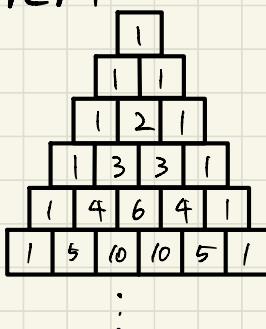
        if (j == 0 || j == N)

            arr[i][j] = 1;

        arr[i][j] = arr[i-1][j] + arr[i-1][j-1];

}

}



: