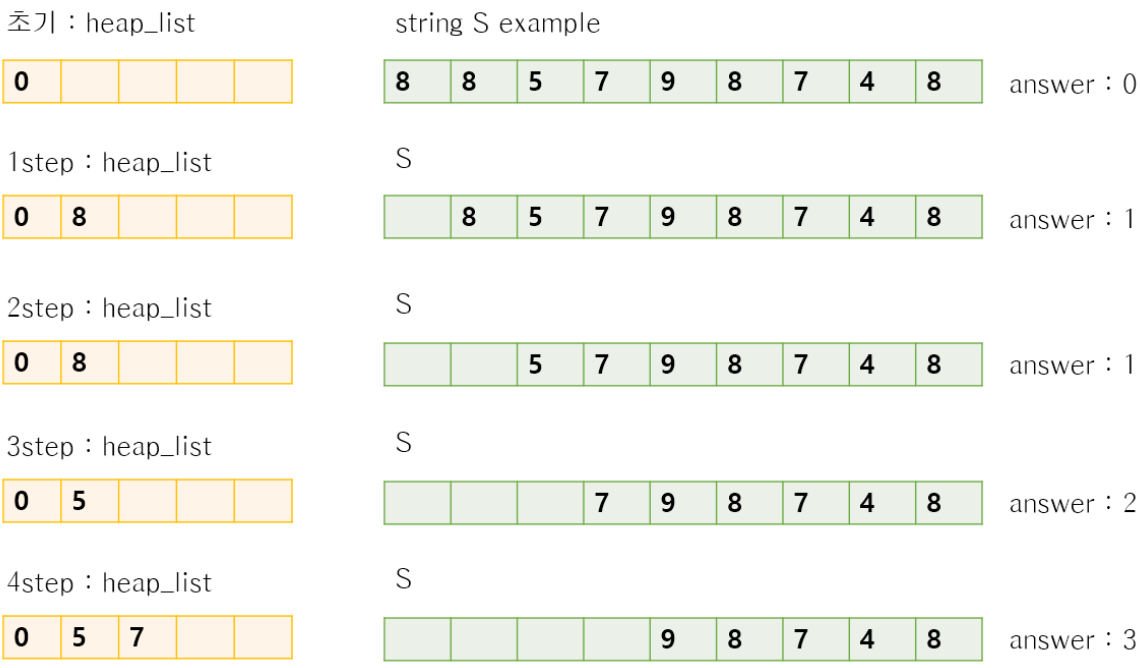
StoneWall

```
def solution(H):
    # write your code in Python 3.6
    heap_list = [0]
    answer = 0
    for h in H:
        while(True):
            if heap_list[-1] > h: heap_list.pop()
            else : break
        if heap_list[-1] == h: pass
        elif heap_list[-1] < h :
            answer += 1
            heap_list.append(h)
    return answer
```

H의 h가 heap_list의 top(고려해야 할 숫자 중 가장 큰 수)보다 **클 경우** 하나의 정사각형 블록이 더 필요해진다. 길이가 **같은 경우**는 이미 그 크기의 블록이 있기에 상관없다.(pass) 그리고 h가 heap_list의 top보다 **작을 경우**에는 heap_list의 값들 중 h보다 큰 값은 이미 블록을 만들었기에, h보다 큰 heap_list의 top값은 제거한다.



5step : heap_list

0	5	7	9	

S

					8	7	4	8

answer : 4

6step : heap_list

0	5	7	8	

S

						7	4	8

answer : 5

7step : heap_list

0	5	7		

S

							4	8

answer : 5

8step : heap_list

0	4			

S

								8

answer : 6

9step : heap_list

0	4	8		

S

answer : 7

y축을 기준으로 선을 그어 응용할 때(직사각형을 만들 때) 사용할 수 있는 알고리즘