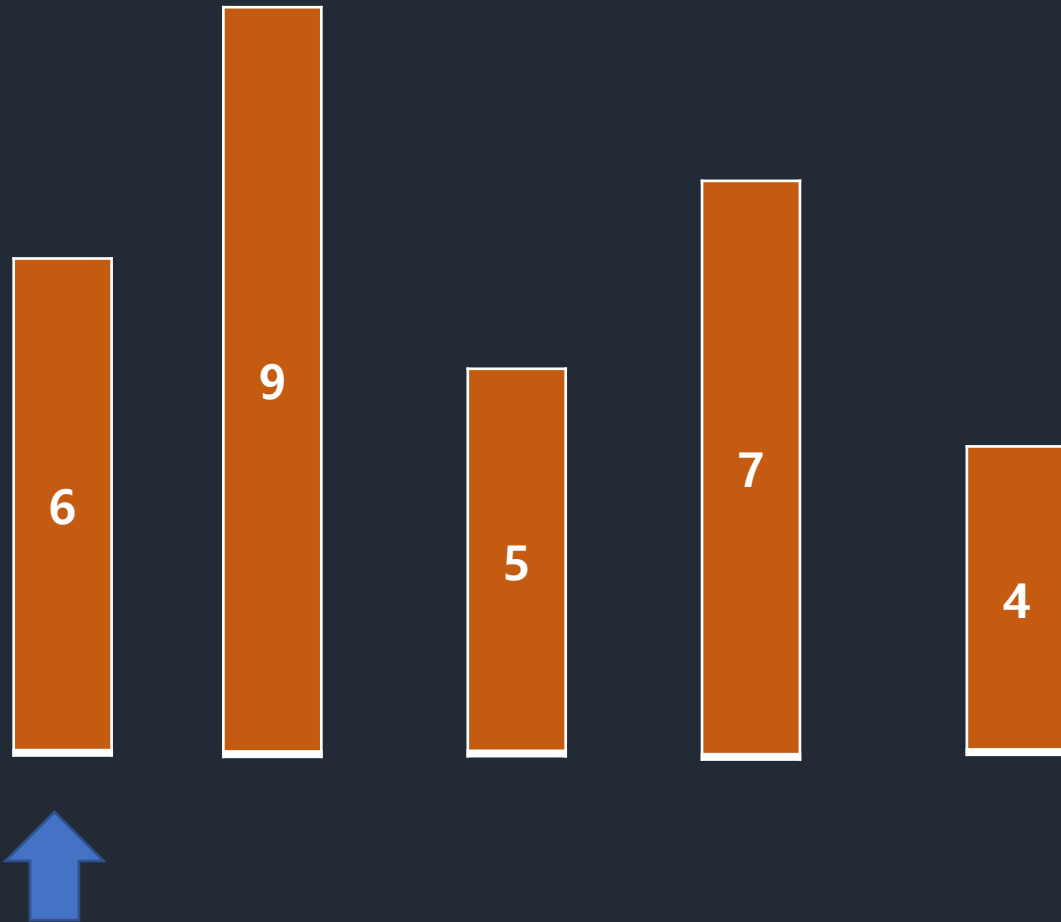


● 문제 풀이

- 이 문제는 스택의 특성을 잘 이해하는데 좋은 문제입니다.
 - 구현 시 고려해야 할 점이 총 2가지 있습니다.
1. 현재 탑의 신호를 받게 될 탑의 높이를 구해야 함.
 2. 신호를 받게 될 탑이 왼쪽으로부터 몇 번째에 위치해 있는지를 알아야 함.
 3. 일반적인 배열로 구현을 하면 시간초과가 뜹니다. <- 이 점을 고려해야 하지만, 바로 아래 설명에서는 설명의 효율을 위해 일반적인 배열로 설명을 하고, 이 부분은 후반부에 설명하겠습니다.

위 두개를 중심으로 프로세스를 설명하겠습니다.

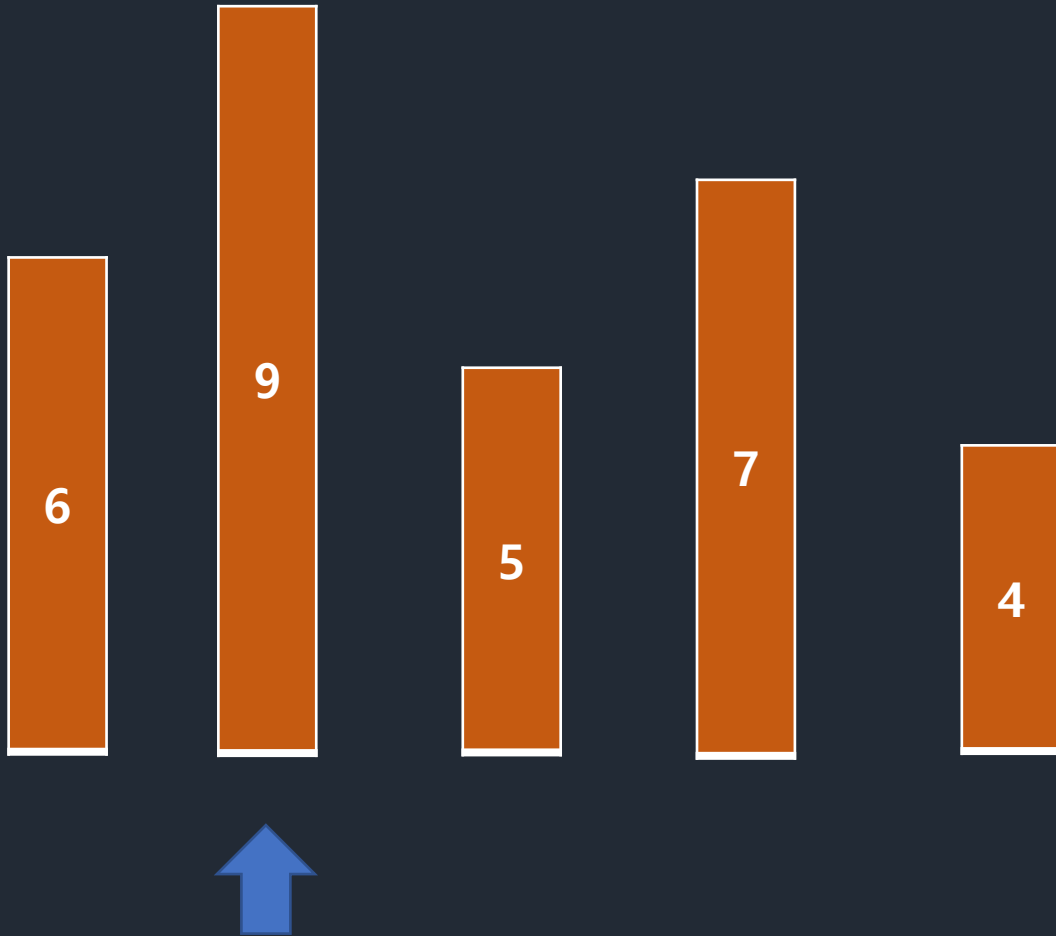
- 1번 탑의 경우, 왼쪽에 아무 것도 없으므로, 신호를 수신할 수 있는 탑이 없습니다.
- 스택에 넣어주고, 2번 탑을 주목합니다.



STACK = [6,

Out : 0,

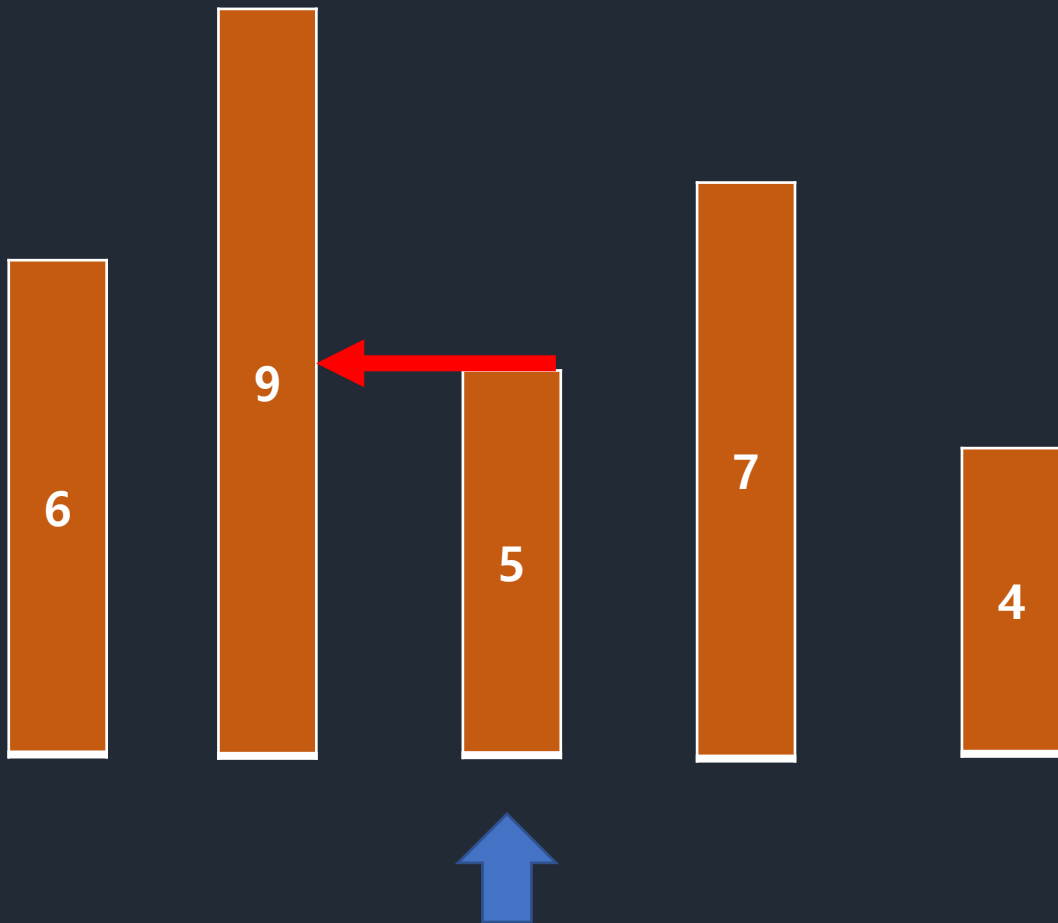
- 2번 탑의 경우, STACK에 있는 크기 6의 탑보다 크므로 신호를 수신할 수 있는 탑이 없습니다.
- STACK에는 1번탑의 6이 들어가 있지만, 2번 탑이 더 크기때문에 앞으로 신호를 받을 일이 없을 것 같습니다. 0을 출력하고, STACK에서 POP해주고 9를 넣습니다.



STACK = [9

Out : 0, 0

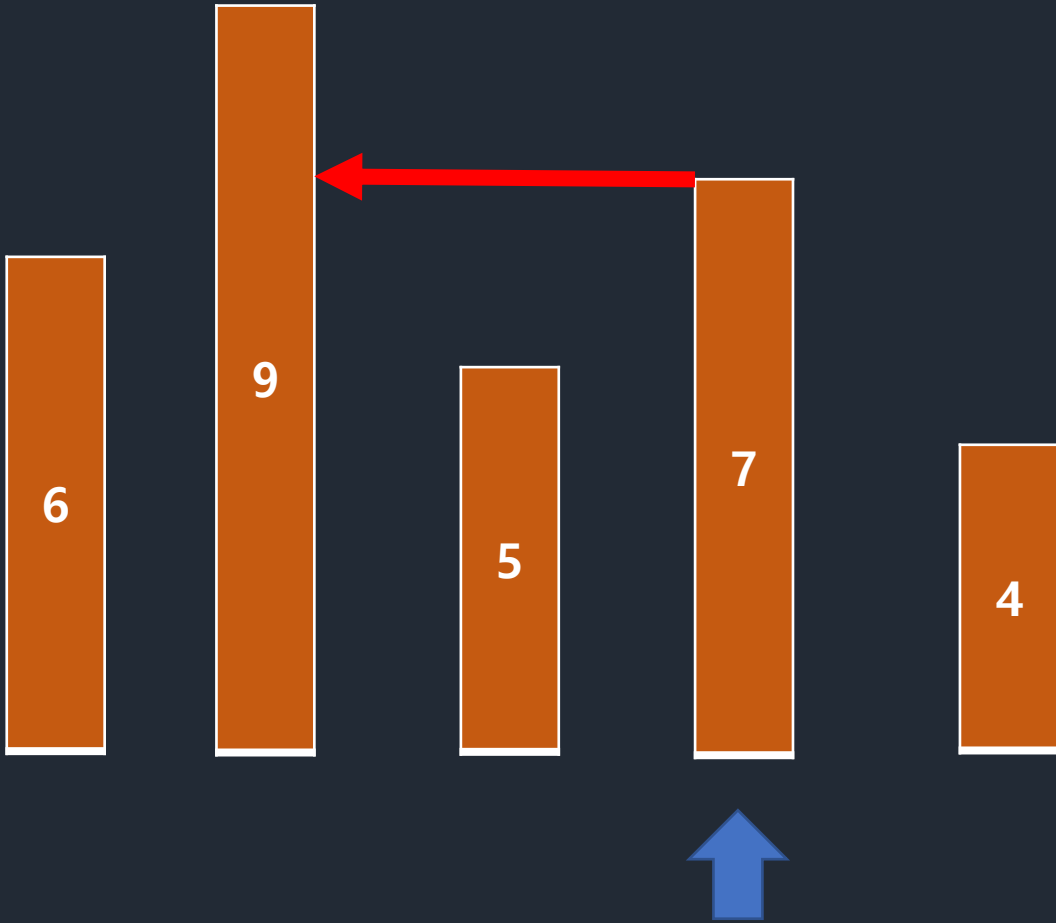
- 3번 탑의 경우, STACK에 있는 크기 9의 탑보다 작으므로 크기 9의 탑이 신호를 수신할 수 있습니다.
- STACK에 3번 탑의 크기를 넣어주고, 크기 9의 탑이 몇 번째인지 **input**으로 받은 배열에서 순회를 하며 찾습니다. 문제에서는 순회를 하면 시간초과가 뜨기 때문에, 후반부에 자세히 설명하겠습니다.



STACK = [9, 5

Out : 0, 0, 2

- 4번 탑의 경우, STACK에 있는 3번의 크기 5의 탑보다 크므로 크기 5의 탑이 신호를 수신할 수 없습니다.
- STACK에서 POP해주고, 다시 비교를 합니다.
- 이번엔 크기 9의 탑보다 4번 탑의 크기가 작아서 수신이 가능할 것 같습니다.
- 크기 9의 탑의 위치를 찾아 출력해줍니다. 그리고 크기 7의 탑을 push 해줍니다.



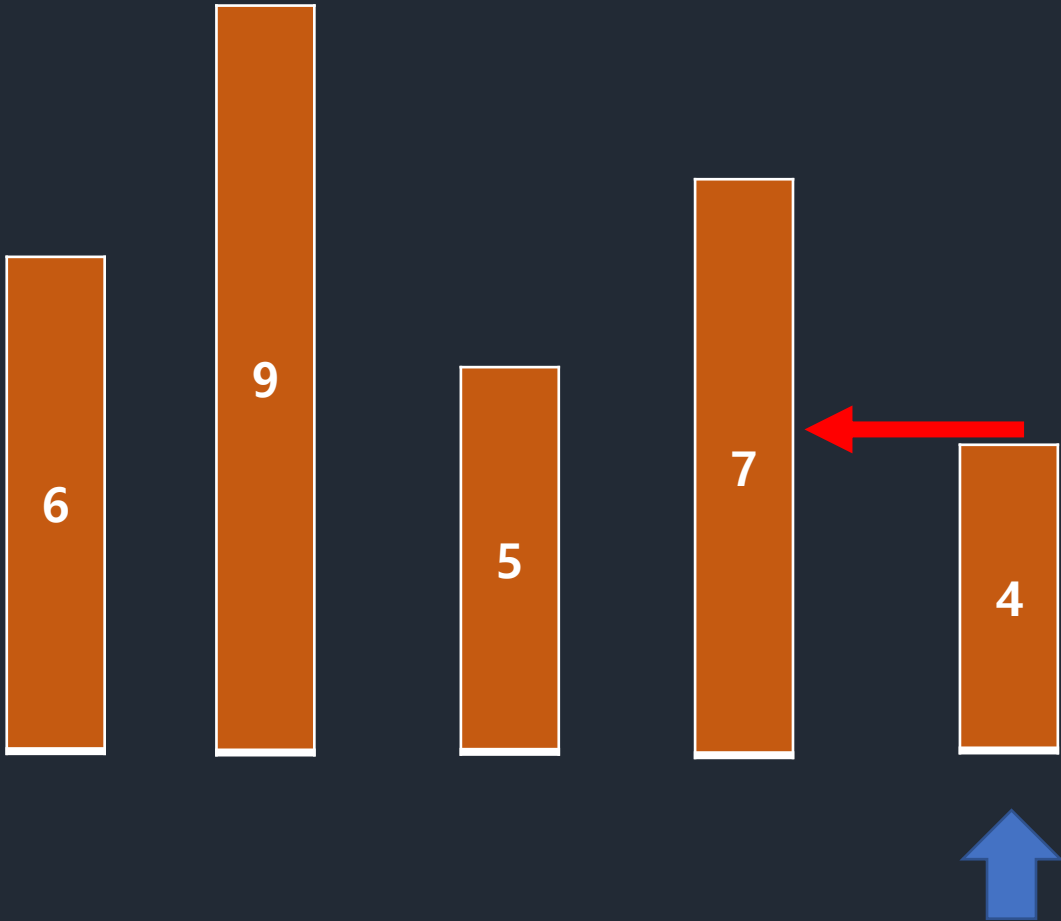
STACK = [9, 5

STACK = [9,

STACK = [7,

Out : 0, 0, 2, 2

- 5번 탑의 경우, STACK에 있는 4번의 크기 7의 탑보다 작으므로 크기 7의 탑이 신호를 수신할 수 있습니다.
- STACK에 PUSH 해주고, 크기 7의 탑의 위치를 순회해서 출력해줍니다.



STACK = [7,
STACK = [7, 4

Out : 0, 0, 2, 2, 4

- 크기 N의 탑의 위치를 배열에서 찾는 방법.

시간복잡도 $O(1)$ 에 해결가능한 방법이 2가지 있습니다. 하지만, 첫 번째 방법은 메모리 초과로 해당 문제에서는 사용이 불가능했습니다. 혹시 모르기 때문에, 두 방법 모두 소개하겠습니다.

1. 배열 index에 직접 접근해서 값을 가져오는 방법.
2. 해시테이블 or 파이썬 딕셔너리를 사용하는 방법.

1번 프로세스

배열 idx에 직접 접근해서 값을 가져오는 방법은 [6,9,5,7,4]의 값을 받아올 때,
Arr = [0,0,0,0,4,3,1,4,0,2] 형태로 저장하는 방법입니다.

즉, 탑 6의 위치정보를 알고 싶을 때, Arr의 6번째 idx를 참고하면(Arr[6]) 위치정보가 1로 저장되어 있는 것을 알 수 있습니다.

코드는 다음과 같습니다.

먼저 tower = [6,9,5,7,4] 안에서 가장 큰 값을 뽑습니다. 여기서는 9가 되겠네요.

배열을 9+1개만큼의 크기로 만들어 줍니다. +1을 하는 이유는 idx를 직접 접근해야 하는데 배열이 0부터 시작하기 때문입니다.

```
arr = [0,0,0,0,0,0,0,0,0,0]
```

이제 for 문을 돌리면서 값을 채워주면 됩니다.

```
for i in range(len(tower)):
    arr[tower[i]] = i
```

2번 프로세스

두 번째 방법은 해시테이블 or 딕셔너리를 사용하는 방법입니다.

이 방법은 키값을 해시값으로 변환하는 연산이 필요하기 때문에, 위 방법보다는 느리지만, 메모리 초과가 뜨는 경우에는 어쩔 수 없이 이 방법을 사용해야 합니다.