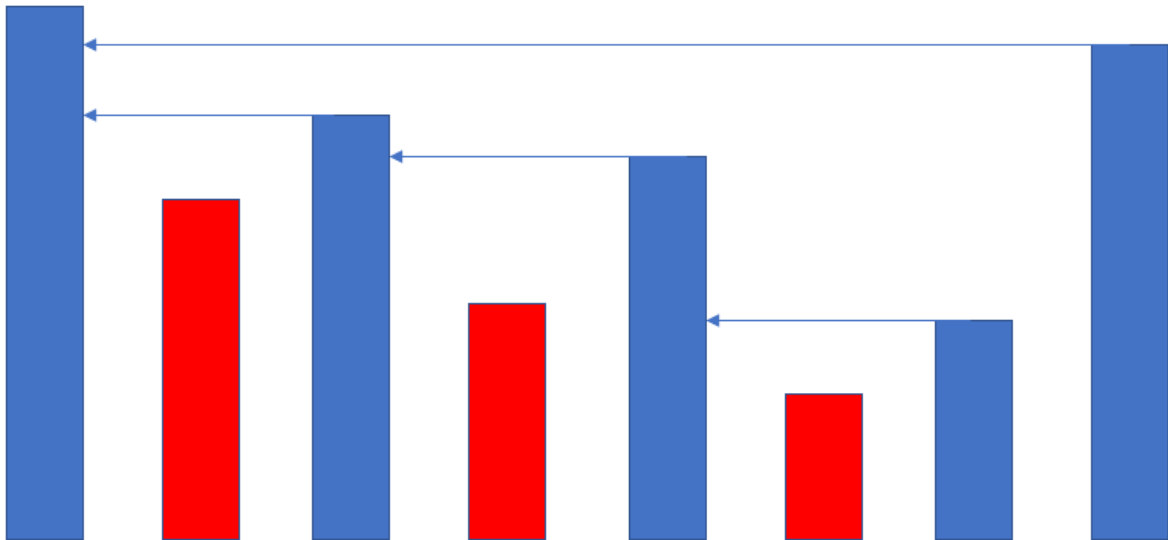


답

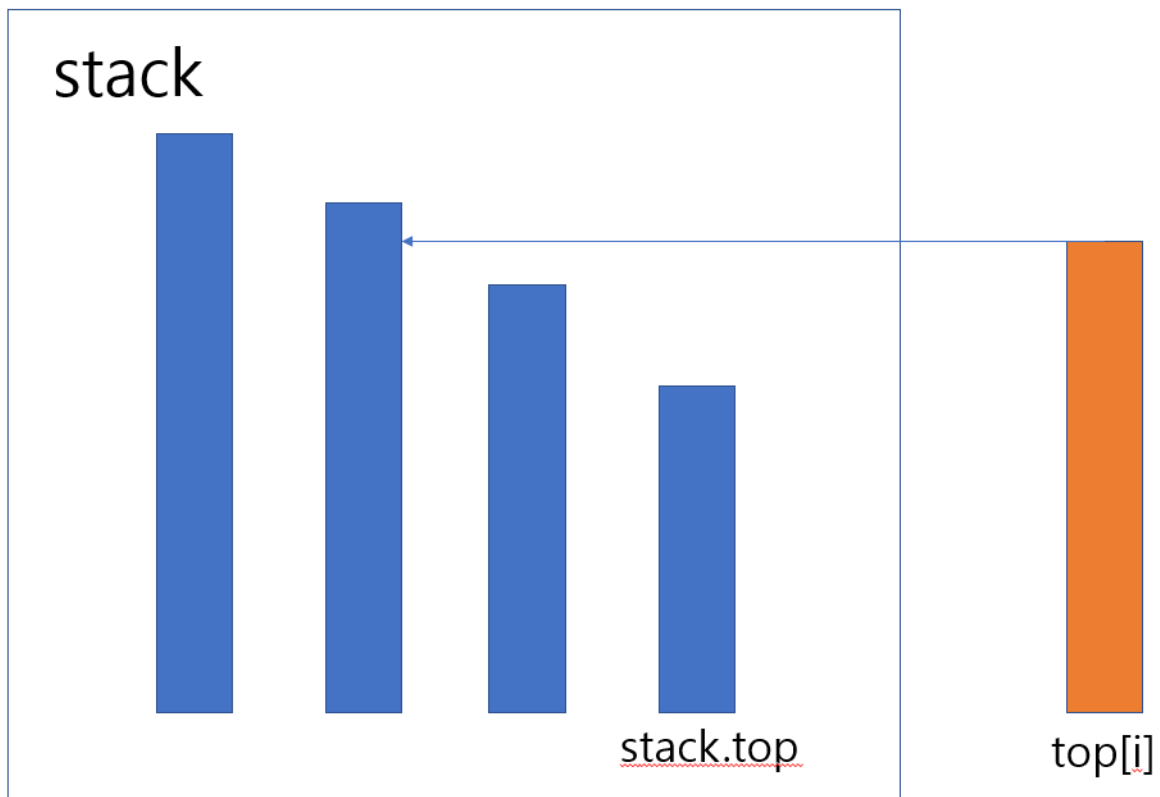
문제 풀이

- 스택 특집이므로 스택이용.
- 아래의 그림을 보면 빨간색 top들은 신호를 못 받는다.
즉, $top[\text{빨간색}] \text{ 높이} < top[\text{빨간색} + 1 \sim @] \text{ 높이}$ 이면 신호를 못 받는다는 것.
=> 빨간색 탑들은 stack에 저장할 필요가 없음!



-----다음페이지 계속-----

- 아래의 그림처럼 stack에는 내림차순으로 탑들이 저장될것이다
stack.top > top[i] 인경우 신호를 받는 것이므로 그대로 cout << stack.top하면 되고
stack.top < top[i] 인경우 stack.pop()을 하여 stack.top > top[i]인 탑을 찾아내서 출력하면 된다. Stack이 완전 비게 되면 top[i]의 신호를 받을 탑은 없으므로 cout << 0을 출력하고 top[i]를 스택에 push한다.



괄호

문제풀이

- VPS 조건1: (로 시작하여)로 쌍을 이루며 끝나야한다.
- VPS 조건2: '('+VPS1+')' 또한 새로운 VPS이고
VPS1+VPS2 또한 새로운 VPS이다.
- '('일 때 +1, ')'일 때 -1로 하여 문자열을 끝까지 읽어드렸을 때

숫자가 0으로 끝나면 YES, 0이 아니면 NO

//주의할 점: ')'시작하여 '('로 끝나면 안된다. 이것은 `str[i] == ')'`일 때
`COUNT <= 0` 이하이면 ')'로 시작한 것이기 때문에 "NO"출력하게 해주면 해결

스택 수열

오후 시간초과 나서 블로그에 있는 코드 도움 받아서 성공.

문제풀이

- Stack 사용
- 0. 현재 숫자를 0으로 지정하고, 숫자를 읽어드린다.
(stack에 push할 때 마다 '+'을 정답을 저장하는 vector에 저장.)
(stack에 pop할 때 마다 '-'을 정답을 저장하는 vector에 저장.)
- 1. 현재 숫자 < 읽어드린 숫자이면 현재 숫자 < N <= 읽어드린 숫자까지 stack에 push한다. 그리고 stack.top을 pop()한다.
현재 숫자는 읽어드린 숫자가 된다.
(pop했다고 해서 현재 숫자가 읽어드린 숫자 - 1 아님)
- 2. 현재 숫자 > 읽어드린 숫자이면 stack.top을 비교하여
 - 2-1. Stack.top == 읽어드린 숫자이면 그대로 pop()한다.
 - 2-2. stack.top != 읽어드린 숫자이면 수열이 성립하지 않은 것이기 때문에 "NO"를 출력하고 종료.
- 3. 이런 식으로 N까지 반복하고 "NO"조건을 만나지 않으면
+,- 연산 순서가 저장되어 있는 vector 출력.