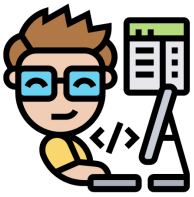


DAILY PROGRAMMING CHALLENGE

WISDOM SPROUTS
IT Training Hub

Sort a Stack Using Recursion

Given a stack of integers, your task is to write a function that sorts the stack in ascending order. You are not allowed to use any additional data structure like arrays, lists, or another stack. The only operations you are allowed to perform are push, pop, and peek on the stack. The sorting must be performed using recursion only.

You need to recursively sort the stack, ensuring that after the sorting process, the stack remains sorted without using any additional data structures. You can only use the stack's own operations and recursion to solve this problem.

Input:

- A stack containing integers. The stack may have positive, negative, or zero values.

Output:

- The input stack should be sorted in ascending order (smallest elements on the top and largest at the bottom) after the sorting operation is performed.

Examples:

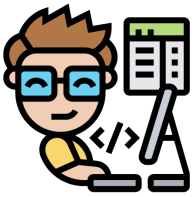
- Example 1
  Input: [3, 1, 4, 2]
  Output: [1, 2, 3, 4]
  Explanation:
  - Initially, the stack contains [3, 1, 4, 2].
  - After sorting using recursion, the stack becomes [1, 2, 3, 4].

Constraints:

- You are not allowed to use any additional data structure such as an array, list, or another stack.
- You can only use stack operations (push, pop, peek) and recursion.
- The input stack can have up to 10^4 elements.

Test Cases:

1. Input: [7, 1, 5]



Output: [1, 5, 7]

2. Input: [5]
   Output: [5]
3. Input: [-3, 14, 8, 2]
   Output: [-3, 2, 8, 14]
4. Input: []
   Output: []
5. Input: [3, 3, 3]
   Output: [3, 3, 3]

Edge Cases:

1. Single Element: If the stack contains only one element, it is already sorted.
2. All Elements are the Same: If all elements in the stack are the same, no changes are required.
3. Negative Numbers: The stack may contain negative integers, and they should also be sorted in ascending order.
4. Empty Stack: If the stack is empty, no operation is needed.