**Sort a stack**

Given a stack, the task is to sort it such that the top of the stack has the greatest element.

**Example 1:**

**Input:**

Stack: 3 2 1

**Output:** 3 2 1

**Example 2:**

**Input:**

Stack: 11 2 32 3 41

**Output:** 41 32 11 3 2

**Your Task:**  
You don't have to read input or print anything. Your task is to complete the function **sort()**which sorts the elements present in the given stack. (The sorted stack is printed by the driver's code by popping the elements of the stack.)

**Expected Time Complexity**: O(N\*N)  
**Expected Auxilliary Space**: O(N) recursive.

**Constraints:**  
1<=N<=100  
  
**Note:**The **Input/Ouput** format and **Example** given are used for system's internal purpose, and should be used by a user for **Expected Output** only. As it is a function problem, hence a user should not read any input from stdin/console. The task is to complete the function specified, and not to write the full code.