Sort a stack using recursion

import java.util.\*;

public class MyClass {

public static void main(String args[])

{

Stack<Integer> stack=new Stack<Integer>();

stack.push(5);

stack.push(1);

stack.push(0);

stack.push(2);

sort(stack);

print(stack);

}

public static void sort(Stack stack)

{

if(stack.size()==1)

{

return;

}

int temp=stack.peek();

stack.pop();

insert(stack,temp);

}

public static void insert(Stack stack,int temp)

{

if(stack.size()==0 || stack.peek()<=temp)

{

stack.push(temp);

return;

}

int temp1=stack.peek();

stack.pop();

insert(stack,temp);

stack.push(temp1);

}

public String print(Stack stack)

{

String s="";

while(!stack.isEmpty())

{

int temp2=stack.peek();

stack.pop();

s=s+ String.valueOf(temp2);

}

return s;

}

}

https://leetcode.com/problems/implement-queue-using-stacks/