Assignment: Counting Recursive Calls using Stack

January 2024 CSE 106 Online B1-B2

Time: 30 minutes

Objective

In this assignment, you are required to determine how many times the Fibonacci function `fib(n)` is called during its execution. You will solve this problem by simulating the recursive calls using a stack data structure.

Problem Description

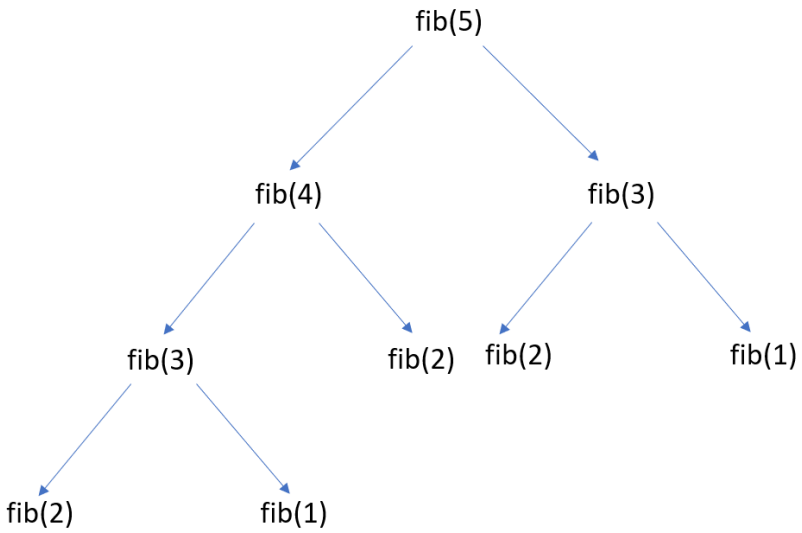
The Fibonacci function is defined as:

$$\text{fib}(n) = \begin{cases} 1 & \text{if } n \leq 2 \\ \text{fib}(n-1) + \text{fib}(n-2) & \text{if } n > 2 \end{cases}$$

Your task is to calculate how many times the function `fib` is called for a given input `n` using a stack-based approach.

Example

For the input `n = 5`, the Fibonacci function calls would be:



In total, the function `fib` is called 9 times.

Example Output

Here is an example of the expected behavior:

Sample Input

```
4
5
8
9
10
```

Sample Output

```
-------------Testing Fibonacci Call Counting-------------
Test Case 1:
The Fibonacci function is called 9 times for n = 5
--------------------
Test Case 2:
The Fibonacci function is called 41 times for n = 8
--------------------
Test Case 3:
The Fibonacci function is called 67 times for n = 9
--------------------
Test Case 4:
The Fibonacci function is called 109 times for n = 10
--------------------
```

Requirements

You need to implement the solution using only stack data structure and stack related operations e.g. push, pop and top. You are not allowed to use recursion or any other data structures like arrays.

Hints

- Use both implementations (**StackArray.h** and **StackLinkedList.h**) to solve the problem. You have been given a skeleton code **online_b1b2.cpp** where you need to include the corresponding header files and implement the **countFibCalls** function. You do not have to worry about output formatting. The skeleton will do that for you. Inputs will be in `input.txt` and outputs will be in `output.txt`.
- For `n=5` the stack contents could be like below stepwise

```
[5]
[4,3]
[4,2,1]
[4]
[3,2]
[3]
[2,1]
[]
```