Problem 3: Algorithm and Pseudocode

Saraswati Saud
Student ID: 40115097

1 Introduction

Following are the algorithms and the pseudo-codes of function F4, $\Gamma(x)$:

Algorithm 1 Recursive Approach - $\Gamma(x)$

```
procedure $functionF4(x)$
    in: double x
    out: double result
    if (x < 0) then return "Wrong Input"
    else
        if (x == x.5) then
            result = halfFactInteger(x)
        else
            result = factInteger(x)
        return result

procedure $halfFactInteger(x)$
    in: double x
    out: double result
    if (($x == 0.5$) then
        return 1.77
    else
        return $(x - 1) * halfFactInteger(x - 1)$

procedure $factInteger(x)$
    in: double x
    out: double result
    if (($x == 1$) then
        return 1
    else
        return $(x - 1) * factInteger(x - 1)$
```

Algorithm 2 Iterative Approach - $\Gamma(x)$

```
procedure functionF4(x)
    in:  double x
    out:  double result
    if (x < 0) then return ”Wrong Input”
    else
        if (x == x.5) then
            result = halfFactInteger(x)
        else
            result = factInteger(x)
        return result

procedure halfFactInteger(x)
    in:  double x
    out:  double result

    double fact = 1.77
    double result = 1.0

    for (int i = 1; i < x; i++) do
        double value = x - i
        result *= value
    return fact * result

procedure factInteger(x)
    in:  double x
    out:  double result

    double fact = 1.0

    for (int i = 1; i < x; i++) do
        fact = fact * i
    return fact
```

2 Algorithm Description

2.1 Algorithm 1

Following are the details of algorithm 1:
Time Complexity: O(n)
Space Complexity: O(n)
Approach: Recursion

Advantages

- Reduces time complexity.
- Adds clarity and reduces the time needed to write and debug code.
- Reduces unnecessary calling of function and length of a code.

Disadvantages

- Recursion is usually slower due to the overhead of maintaining the stack.
- It usually uses more memory for the stack.
- Recursive methods will often throw a StackOverflowException when processing big sets.

2.2 Algorithm 2

Following are the details of algorithm 2:
Time Complexity: O(n)
Space Complexity: O(1)
Approach: Iterative

Advantages

- Algorithm 2 does not suffer from stack overflow because all operations are conducted on the heap.
- The space complexity of Algorithm 2 is O(1).

Disadvantages

- An infinite loop for iteration occurs when the condition never fails.
- Not efficient for larger inputs as it requires more time to execute.

3 Conclusion

Algorithm 1 has greater space requirements than Algorithm 2 as all the functions will remain in the stack until the base case is reached. In addition to this, algorithm 1 (i.e. recursive approach) has greater time requirements because of function calls and returns overhead. Therefore, iterative algorithm is preferred over recursive approach.

References

[1] Medium: Recursion - Pros and Corns
`https://medium.com/@williambdale/recursion-the-pros-and-cons-76d32d75973a`

[2] Geeksforgeeks: Recursive Function,
`https://www.geeksforgeeks.org/recursion/`