Problem 3 - SOEN 6011

Pavit Srivatsan

August 2021

Pseudocode and Algorithm

Calculate: $f(x, y) = x^y$

ALGORITHM 1: Iterative algorithm to calculate x^y

1. function **power_function_iterative(x,y)**
in: double number x,y
out: double number result
2. $result \leftarrow 1$
3. $temp \leftarrow 1$
4. for $temp \leq y$ do
5. $\quad result \leftarrow result * x$
6. $\quad temp \leftarrow temp + 1$
7. end for
8. return result

The output is stored in result, which is initially set to 1. It is then looped from 1 to y, x number of times, incremented by one on each iteration and on each iteration we multiply result by x. At the end of the loop value of result is equal to x^y.

ALGORITHM 2: Recursive Divide and Conquer algorithm to calculate x^y

1. function **power_function_recursive(x,y)**
in: double number x,y
out: double number result
3. $power \leftarrow exponent\_helper(x, y)$
4. $result = power$
5. return result

1. function **exponent_helper(x,y)**
in: double number x, y
out: double number sum
2. if $x < 0$ then
3. $x \leftarrow 1.0/x$
4. $y \leftarrow -y$
5. return $exponent\_helper(x, y)$
6. else if $y = 0$ then
7. return 1.0
8. else if $y = 1$ then
9. return x
10. else if $y \mod 2 = 0$ then
11. $y \leftarrow y * y$
12. $y \leftarrow y/2$
13. return $exponent\_helper(x, y)$
14. else
15. $x \leftarrow x * x$
16. $y \leftarrow y - 1$
17. $y \leftarrow y/2$
18. return $exponent\_helper(x, y)$
19. end if

A helper function called exponent_helper is defined which calculates x^y. In the base case when $y = 0$, we return 1, otherwise when $y = 1$ we return x. When x is even we recurse on $x = x * x$ and $y = y/2$. In case when x is odd we recurse on $x = x * x$ and $y = (y - 1)/2$. In the end, in our main function power_function_recursive we multiply the result of exponent_helper to the value of a and return our result.

Advantages and Disadvantages

Algorithm 1:

Advantages:
1. In terms of space complexity, iterative algorithms don't suffer from stack overflow because all operations are done on the heap.
2. They are easy to comprehend by humans and have better readability.
Disadvantage:
1. The time complexity of the iterative algorithm is $O(n)$, hence it is not very efficient for larger inputs in terms of time.
2. Proper terminating condition for loop is important or else it might lead to infinite looping.

Algorithm 2:

Advantages:
1. The time complexity of the recursive algorithm is $O(logn)$. This recursive algorithm is optimized (tail recursive) so that we don't get stack overflow error and it handles large inputs better.
2. Recursion has higher maintainability than looping. Handling the base case properly requires little or no modifications.
Disadvantages:
1. Due to the continuous allocation of memory space leading to a stack overflow, efficiency is significantly affected.
2. It is difficult to comprehend. A certain level of expertise is required for understanding it vividly.

Conclusion

Recursive algorithm which is based on Divide and Conquer Algorithmic Strategy is preferred over iterative algorithm

References

[1] TutorialsPoint,
`https://www.tutorialspoint.com/java/lang/math_pow.htm`

[2] GeeksforGeeks,
`https://www.geeksforgeeks.org/write-a-c-program-to-calculate-powxn/`

[3] MathBitsNotebook,
`https://mathbitsnotebook.com/Algebra1/FunctionGraphs/FNGTypeExponential.html`