Problem 3 - Sinh(x) Algorithms

Jesus Onofre Diaz - 40035954

July 19, 2019

1 Function $Sinh(x)$ Recap

The hyperbolic function Sinh of a given number x is denoted by the expression $sinh(x) = \frac{e^x - e^{-x}}{2}$ where e often called Euler's number is an irrational number which is usually calculated using the expression $e = (\frac{1+1}{n})^n$ where n is any natural number, the greater the n number is the closer to most commonly used value by this constant which is 2.71827.

1.1 Algorithms

The algorithm for calculating the $sinh(x)$ involves the implementation of different procedures such as the calculation of the e value which can be calculated using different approaches, the power of a number n^n, the power of a number n^{-n} and the calculation of the absolute value of a number $\mid n \mid$ where $n \in R$.
The procedure for calculating e was implemented using two different approaches, the first one is by resolving the expression $e = (\frac{1+1}{n})^n$. The second approached used for calculating the e value is by resolving the equation $e = (\frac{1}{0!} + \frac{1}{1!} + ... \frac{1}{n!})$ where ! is the factorial of n and n is any natural number.

The procedure for calculating the positive power of n was implemented using a recursive function and an iterative function.
The procedure for calculating the negative power of n was implemented using the expression $a^{-n} = \frac{1}{a^n}$.
The procedure for calculating the absolute value of $-n$ was implemented using the expression $abs(-n) = -n * -1$.

1.2 Algorithm 1 Advantages

The algorithm 1 uses a simple implementation of its involved functions which allows a better usage of the resources such as memory and processor, plus this algorithm applies the principle of Modularity which helps in the Maintainability of the system and in the Readability of the program.

1.3 Algorithm 1 Disadvantages

Some operations are set to a finite number of operations and some results could vary slightly when numbers are extreme large.

1.4 Algorithm 2 Advantages

The algorithm 2 is implemented applying the principle of separation of concerns, all the sub modules in the function are highly cohesive and low coupled, this allows the program being maintainable and change prone.

1.5 Algorithm 2 Disadvantages

The use of some recursive and iterative methods could demand more resources and make the application to have low performance when dealing with a great load of operations.

Algorithm 1 Calculate $sinh(x)$ Function

```
Require: value: $x from - \propto to + \propto$                    ▷ where $x \in R$
Ensure: $result = \frac{e^x - e^{-x}}{2}$
procedure CalculatePower($base$, $exponent$)
    $power \leftarrow 1$
    for $i \leftarrow 1, exponent$ do
        $power \leftarrow power * base$
    end for
    return $power$                ▷ returns the power of a positive exponent
end procedure

procedure CalculateNegativePower($x$)
    $power \leftarrow 0$
    $power \leftarrow \frac{1}{x}$
    return $power$                ▷ returns the power of a negative exponent
end procedure

procedure AbsoluteValue($x$)
    $a \leftarrow x * -1$
    return $a$                              ▷ returns the absolute value of x
end procedure

procedure CalculateEuler( )
    $value \leftarrow 5000$                            ▷ value is a set number
    $e \leftarrow 0$
    $calcule_e \leftarrow 0$
    $e \leftarrow \frac{1+1}{value}$
    $calcule_e \leftarrow$ CalculatePower($e, value$)
    return $calcule_e$                                   ▷ returns euler value
end procedure

procedure CalculateSinhX($x$)
    $aux1 \leftarrow 0$                    ▷ receives the value of positive power
    $aux2 \leftarrow 0$                    ▷ receives the value of negative power
    $aux1 \leftarrow$ CalculatePower($CalculateEuler, x$)
    $aux2 \leftarrow$ CalculateNegativePower($aux1$)
    return $(aux1 - aux2) * 0.5$                           ▷ returns sinh of x
end procedure

$result \leftarrow$ CalculateSinhX($x$)                  ▷ Final result of $sinhx$
```

Algorithm 2 Calculate $sinh(x)$ Function, second approach for e and power

```
Require: value: $x from - \propto to + \propto$        ▷ where $x \in R$
Ensure: $result = \frac{e^x - e^{-x}}{2}$
procedure CalculatePower($base$, $exponent$)
    if $exponent \leftarrow 0$ then
        return 1                                    ▷ case base of the recursive function
    end if
    return $base * $ CalculatePower($n, exponent - 1$)      ▷ recursive execution
end procedure

procedure CalculateNegativePower($n, x$)
    $aux \leftarrow n$
    $x \leftarrow$ AbsoluteValue($x$)
    for $i \leftarrow 2, x$ do
        $aux \leftarrow aux * n$
    end for
    return $response$                ▷ returns the power of a negative exponent
end procedure

procedure AbsoluteValue($x$)
    $a \leftarrow x * -1$
    return $a$                                  ▷ returns the absolute value of x
end procedure

procedure CalculateEulerFactorial($n$ )
    $aux \leftarrow 1$                                      ▷ auxiliar variable
    for $i \leftarrow 1, n$ do
        $aux \leftarrow aux * i$
    end for
    return $aux$                                     ▷ returns euler factorial
end procedure

procedure CalculateEuler( )
    $aux \leftarrow 1$                                      ▷ auxiliar variable
    for $i \leftarrow 1, 10$ do
        $aux \leftarrow aux + (\frac{1}{\text{CalculateEulerFactorial}(x)})$
    end for
    return $aux$                                         ▷ returns euler value
end procedure

procedure CalculateSinhX($x$)
    $euler1 \leftarrow$ CalculatePower($CalculateEuler, x$)
    $euler2 \leftarrow$ CalculateNegativePower($euler1, x$)
    return $(euler1 - euler2) * 0.5$                        ▷ returns sinh of x
end procedure

$result \leftarrow$ CalculateSinhX($x$)                    ▷ Final result of $sinhx$
```