Function:- arccos(x)

Problem-3

Birva Shah (Student ID: 40070973)

19 July 2019

1 Description

There are 2 ways to find arccos(x). One is iterative approach and the other is recursive approach. Taylor's series for the evaluation has been used.

$$\arccos x = \frac{\pi}{2} - \sum_{n=0}^{\infty} \frac{(2n)!}{2^{2n}(n!)^2} \frac{x^{2n+1}}{(2n+1)}, |x|<1 \tag{1}$$

Comparison between these two approaches have been showed in this document. The time complexity of both the algorithms is same but it has been observed that iterative approach is better than the recursive one.
The recursive approach has resulted in high memory consumption compared to the other one.

2 Advantages and Disadvantages of Iterative Algorithm

- If implemented during the earlier stages of the development process allows the team to find functional or design related flaws as early as possible.
- Easily adaptable to the ever-changing needs of the project as well as the client.
- It is the best suited for agile organizations and less time is spent on documenting and more on designing to implement iterative model.
- More resources may be required.
- It is not suitable for small projects project.

Algorithm 1 Calculating: $arccos(x)$ using Iterative Algorithm

function $PI()$

1. $pi\_value \leftarrow 0.0$
2. for $k \leq 9999$

   $first \leftarrow power(-1, k)$

   $second \leftarrow (2 * k) + 1$

   $value \leftarrow first/second$

   $pi\_value \leftarrow pi\_value + value$
3. $pi\_value \leftarrow 4 * pi\_value$
4. $returnpi\_value$

function $ARCCOS(x)$

$in : valueofx$

$out : calculatedvalueofarccos(x)inradian$

1. $ans \leftarrow 0$
2. for $n \leq 89$

   $a = factorial(2 * n)$

   $if(Double.isInfinite(a))$

   break

   $b \leftarrow power(2, (2 * n))$

   $c \leftarrow factorial(n)$

   $d \leftarrow power(c, 2)$

   $A \leftarrow (a/(b * d))$

   $exp \leftarrow (2 * n) + 1$

   $e \leftarrow power(num, exp)$

   $B \leftarrow e/exp$

   $AB \leftarrow (A * B)$

   $ans \leftarrow ans + AB$
3. $pivalue \leftarrow pi()$
4. $finalans \leftarrow ((pivalue/2) - ans)$
5. $returnfinalans$

function $POWER(c, j)$

$in : valueofcandj$

$out : valueofpower(c, j)$

1. $ans \leftarrow 1.0$
2. $if(j == 0)$

   $ans \leftarrow 1$

   else

   for $i \leq j$

   $ans \leftarrow c * ans$

3.return ans

function $FACTORIAL(i)$

$in : valueofi$

$out : valueoffactorial(i)$

1. $ans \leftarrow 1.0$
2. $if(i == 0)$

   $ans \leftarrow 1$

   else

   $forj \leq i$

   $ans \leftarrow ans * j$
3. $returnans$

Algorithm 2 Calculating: $arccos(x)$ using Recursive Algorithm

function $PI()$
1. $pi\_value \leftarrow 0.0$
2. for $k \leq 9999$
   $first \leftarrow power(-1, k)$
   $second \leftarrow (2 * k) + 1$
   $value \leftarrow first/second$
   $pi\_value \leftarrow pi\_value + value$
3. $pi\_value \leftarrow 4 * pi\_value$
4. $returnpi\_value$

function $ARCCOS(x)$
$in : valueofx$
$out : calculatedvalueofarccos(x)inradian$
$ans \leftarrow FUNC(x, 0, 0)$
$ans \leftarrow ((PI/2) - ans)$
$returnans$

function $FUNC(value, steps, ans)$
$in : valueofvalue, stepsandans$
$out : valueoffunc(value, steps, ans)$
1. $a = factorial(2 * steps)$
2. $if(Double.isInfinite(a))$
   $stepsByMethod = steps - 1$
   $returnans$
3. $b \leftarrow power(2, (2 * n))$
4.$c \leftarrow factorial(n)$
5.$d \leftarrow power(c, 2)$
6.$A \leftarrow (a/(b * d))$
7. $exp \leftarrow (2 * n) + 1$
8. $e \leftarrow power(num, exp)$
9. $B \leftarrow e/exp$
10. $AB \leftarrow (A * B)$
11. $ans \leftarrow ans + AB$
12.$steps \leftarrow steps + 1$
13. $returnFUNC(value, steps, ans)$

function $POWER(c, j)$
$in : valueofcandj$
$out : valueofpower(c, j)$
1. $ans \leftarrow 1.0$
2. $if(j == 0)$
   $ans \leftarrow 1$
   else
   for $i \leq j$
   $ans \leftarrow c * ans$
3.return ans

function $FACTORIAL(i)$
$in : valueofi$
$out : valueoffactorial(i)$
1. $ans \leftarrow 1.0$
2. $if(i == 0)$
   $ans \leftarrow 1$
   else
   $forj \leq i$
   $ans \leftarrow ans * j$
3. $returnans$