DAA Assignment 3

Priyal Gupta(IHM2017004), Aditya Gupta(IHM2017005),

Pratham Prakash Gupta(IHM2017003) and Rohit Ranjan(IHM2017001)

January 27,2019

1 Abstract

The problem is to determine the value of cosec$\theta, if\ \ sin\theta\ \ is\ \ given.$

We have to solve this problem without using any multiplication, division or modulo operator. There are many algorithms to solve the above problem. We have to select the one which has least time complexity. The approach shown below has four functions for calculating dividend, quotient, multiplication of dividend and quotient and generating new dividend for calculation of cosecant.

2 Keywords

Sin theta: The sine is a trigonometric function of an angle. The sine of an acute angle is defined in the context of a right triangle: for the specified angle, it is the ratio of the length of the side that is opposite that angle to the length of the longest side of the triangle (the hypotenuse).

Cosec theta: The cosecant of an angle is the reciprocal of its sine, that is, the ratio of the length of the hypotenuse to the length of the opposite side.

3 Introduction

3.1 Question

The problem is to determine the value of cosec $\theta, if\ sin\theta\ is\ given\ without\ using\ any\ multiplication,\ division\ or\ modulo\ operator.$

3.2 Idea

We have to develop an algorithm to find cosec theta if sin theta is given. There can be several ways to find the same. Our goal is to develop the most efficient algorithm with the least time complexity.We have to solve this problem without using any multiplication, division or modulo operator. But here the problem is that the value of sine can contain infinite number of decimal places.So it must be first approximated to a number with fixed number of decimal places by storing it in a string. Then 1 should be divided byt the obtained number. The division is done in a way such that the numbers must be in the form of a/b where a and b are integers. a is made integer by removing its decimal digits and putting same number of zeroes in numerator.

4 Algorithm Design

4.1 Part a: Multiplication of divisor and quotient

4.1.1 Variables:

res: A variable to store the product of divisor and quotient.

4.1.2 Function:

min: Function to find minimum of two numbers.
max: Function to find maximum of two numbers.

4.1.3 Pseudo Code:

```
divisorMultiQuotient (divisor,quotient)
    res← 0
    for i← 1 to $min(divisor, quotient)$ do
        res← $res + max(dividor, quotient)$
    return res
end
```

4.2 Part b: Obtaining dividend

4.2.1 Variables:

num: String to store the dividend

4.2.2 Pseudo Code

```
getDividend(number)
    num← "1"
    for i← 0 to $number.length()$ do
        num← $num +$ "0"
    return stoi(num)
end
```

4.3 Part c: Generating Dividend

4.3.1 Pseudo Code

```
GeneratingDividend(divisor,remainder,
        count)
    count← $-1$
    while remainder≤ $divisor$ do
        newRem← 0
        for i← 0 to 10 do
            newRem← $newRem + remainder$
        remainder← $newRem$
        count← $count + 1$
end
```

4.4 Part d: Obtain Quotient

4.4.1 Pseudo Code

```
GetQuotient(dividend,divisor)
    quotient← 0
    while dividend≥ $divisor$ do
        dividend← $dividend - divisor$
        quotient← $quotient + 1$
    return quotient
end
```

4.5 Part e: Main Function

4.5.1 Pseudo Code

```
Main()
    if(number≤ $-1$ or $number \geq 1$)then
            print(Not a valid sine value)
    exit
    if(number≤ 0)then
        ans ← $ans +$ " − "
        number← $number * (-1)$
    stringNumber← $toString(number)$
    stringNumber← $substr(2, stringNumber.len$
    gth-2)
    dividend← $getDividend(stringNumber)$
    divisor← $stoi(stringNumber)$
```

```
quotient← $GetQuotient(dividend, divisor)$
remainder← $dividend - divisorMultiQuotient$(divisor, quotient)
if(remainder= 0) then
    print(ans)
    exit
ans← $ans + "."$
p← $number\ of\ decimal\ places\ in\ str-$ingNumber
for    i← 0 $to$ $p$ do
    cnt← −1
    dividend← $GenerateDividend(divi-$sor, remainder, count)
    for    j← 0 $to$ $count$ do
        ans← $ans + "0"$
    quotient← $GetQuotient(dividend,$ divisor)
    remainder← $dividend - divisorMul-$tiQuotient(divisor, quotient)
    ans← $ans + toString(quotient)$
    if(remainder= 0) then
        break
print(ans) end
```

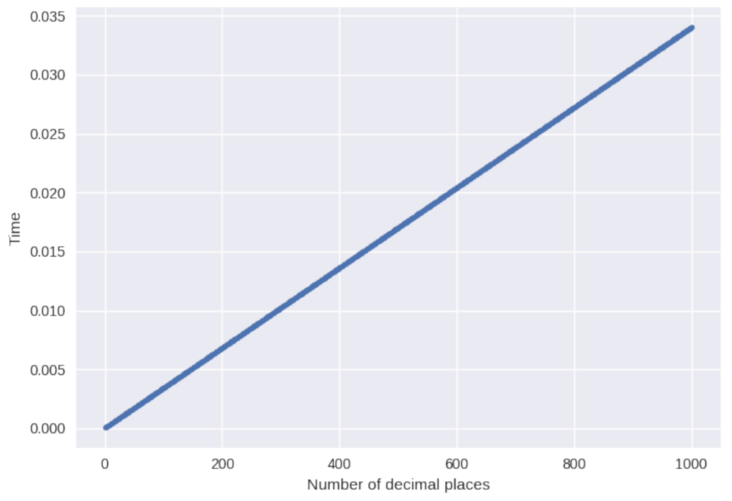
5 Analysis and Discussion

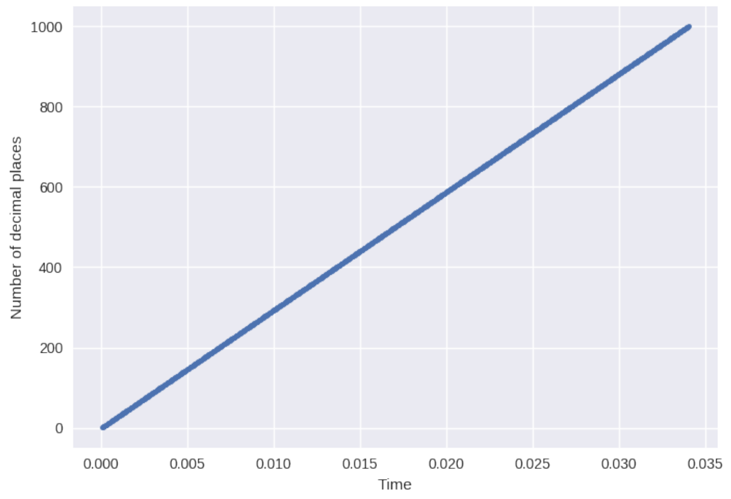
5.1 Time Complexity

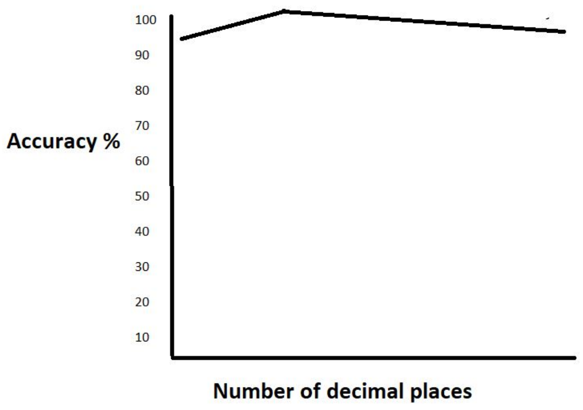
The time complexity is of the form O(min(divisor,quotient) + O(number) + O(log10(divisor)) + O(number of decimal places).
The graph comes out to be of form y=x because one of the above value dominates the other.

Best Case
The best case comes out when the value of sin is such that 1/sin comes out to be perfect integer, i.e. without any decimal values.

Worst Case
The worst case comes out when the value of sin is such that 1/sin is never terminating.







5.2 Space Complexity

The Space Complexity of an algorithm is the maximum amount of space used at any one step. Thus, the space complexity of this algorithm comes out to be O(1).

6 Experimental Setup

Language Used :- C++
Plotting tool :- matplotlib
Report Making Tool :- TextPortable(Latex)

7 Conclusion

We have to develop an algorithm to find cosec theta if sin theta is given.,without using any multiplication, division or modulo operator. There can be several ways to find the same. In the above algorithm, different functions are made for reciprocating the given sine value as direct operations cannot be used.

8 References

Latex :- https://www.sharelatex.com/learn
Code :- https://www.geeksforgeeks.org/multiply-two-numbers-without-using-multiply-division-bitwise-operators-and-no-loops/
Gnu plot :- https://matplotlib.org