SOEN 6011- SOFTWARE ENGINEERING PROCESSES
PROJECT DELIVERABLE 1

Jemish Kishor Paghadar (40080723)
Concordia University, Montreal

July 19 2019

Contents

0.1 Problem 1

0.1.1 Introduction

In mathematics, the logarithm is the inverse function to exponentiation. That means the logarithm of a given number x is the exponent to which another fixed number, the base b, must be raised, to produce that number x.
The exponential function that is $y = b^x$ has the inverse $x = b^y$. So, to express y as a function of x the logarithm was invented by John Napier in 1614.[1]

$F(x) = \log_b x$ is called "x is equal to b to the power y." This is equivalent to saying "y is the base-b logarithm of x."

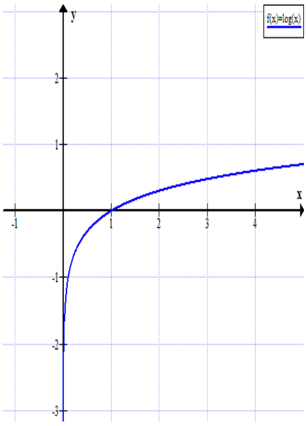
0.1.2 Domain and Co-Domain

- The **domain** is the set of all **positive real numbers**.
- $F(x) = \log_b x$ is not defined for negative values of x, or for 0.
- It is defined for base $b \neq 1$ **and** $b > 0$.
- The **co-domain** is the **set of all real numbers**. (Since the logarithmic function is the inverse of the exponential function, the domain of logarithmic function is the range of exponential function, and vice versa.). This function is continuous and one-to-one

0.1.3 Applications of logarithms

- It is used to simplify multiplication and division by converting these operations into addition and subtraction in **slide rule**. This is done by placing the numbers on a scale which is logarithmic. [1]
- Logs are used in a variety of applications in sciences, some of the most common are: measuring loudness (decibels), measuring earthquake intensity (Richter scale), radioactive decay, and acidity ($pH = -\log_{10}[H^+]$). They are also essential in mathematics to solve certain exponential-type problems.

0.1.4 Properties of logarithms



- The log of a product is the sum of the logs
- The sum of the logs is the log of the products
- The log of a quotient is the difference of the logs
- The difference of the logs is the log of the quotient
- The exponent on the argument is the coefficient of the log
- The coefficient of the log is the exponent on the argument

0.2 Problem 2

0.2.1 Requirements

Requirement : 1

- **ID :** FUNR1
- **TYPE :** Functional Requirement
- **PRIORITY :** 1
- **VERSION :** 1.0
- **DIFFICULTY :** Easy
- **DESCRIPTION :** System shall take two input x and base b and gives base-b logarithm of x. For example, user gives input $x = 10$ and $b = e$ then output will be 1.
- **RATIONALE :** To perform functionality on them.

Requirement : 2

- **ID :** FUNR2
- **TYPE :** Functional Requirement
- **PRIORITY :** 1
- **VERSION :** 1.0
- **DIFFICULTY :** Easy
- **DESCRIPTION :** When user gives negative value or zero as input to x in function $log_b x$, the system shall give the alert message that the result is not possible with this value and it should be within its domain.
- **RATIONALE :** To handle the cases where input is not from its domain.

Requirement : 3

- **ID :** FUNR3
- **TYPE :** Functional Requirement
- **PRIORITY :** 1
- **VERSION :** 1.0
- **DIFFICULTY :** Easy
- **DESCRIPTION :** When any user gives value less than 2 as input to base b in function $log_b x$, the system shall give the alert message that the result is not possible with this value as function is defined for only base $b \neq 1$ and $b > 0$.
- **RATIONALE :** To define the scope of the base b.

Requirement : 4

- **ID :** FUNR4
- **TYPE :** Functional Requirement
- **PRIORITY :** 1
- **VERSION :** 1.0
- **DIFFICULTY :** Easy
- **DESCRIPTION :** When user gives any other data type other than numbers such as string etc. as an input, the system shall not accept that type and show that input type is not properly defined.
- **RATIONALE :** To define that input must be numbers.

Requirement : 5

- **ID :** FUNR5
- **TYPE :** Functional Requirement
- **PRIORITY :** 1

- **VERSION :** 1.0
- **DIFFICULTY :** Easy
- **DESCRIPTION :** When user gives irrational number such as e as input to base b then system shall calculate natural logarithm.
- **RATIONALE :** To define natural logarithm.

Requirement : 6

- **ID :** NFUNR1
- **TYPE :** Non-functional Requirement
- **PRIORITY :** 1
- **VERSION :** 1.0
- **DIFFICULTY :** Easy
- **DESCRIPTION :** The final output shall have some accuracy up to some point.
- **RATIONALE :** To achieve proper accuracy for the system.

Requirement : 7

- **ID :** NFUNR2
- **TYPE :** Non-functional Requirement
- **PRIORITY :** 1
- **VERSION :** 1.0
- **DIFFICULTY :** Easy
- **DESCRIPTION :** The system shall respond in few seconds.
- **RATIONALE :** To achieve response time and performance speed.

0.3 Problem 3

0.3.1 Algorithm 1

The logarithm of a given number x is the exponent to which another fixed number, the base b, must be raised, to produce that number x as the logarithm is the inverse function to exponentiation. There are many different ways to implement logarithm functions such as using Maclaurin's Series, recursion method etc.
The Maclaurin's Series is represented as,

$$f(x) = f(0) + \frac{f'(0)}{1!}x + \frac{f''(0)}{2!}x^2 + \frac{f'''(0)}{3!}x^3 + ...$$

as our function is $f(x) = log_b x$ is calculated using $\frac{\ln x}{\ln b}$, but Maclaurin's series cannot be used to find a series for $\ln x$. So, we use $\ln(l + x)$, which produces finite values for successive derivatives when $x = 0$.

$f(x) = \ln(1 + x)$
$f'(x) = \frac{1}{1+x}$
$f''(x) = -\frac{1}{(1+x)^2}$
$f'''(x) = \frac{2}{(1+x)^3}$

we have $f(0) = 0, f'(0) = 1, f''(0) = -1, f'''(0) = 2$ and Putting all these values in equation, we get

$$\ln(1 + x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} .... \quad (1)$$

Similarly, modified logarithm series can be represented as,

$$\ln(1 - x) = -x - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} .... \quad (2)$$

Instead of Considering above equations, it is better to consider their difference (1)-(2) for obtaining a better convergence.

$$\ln(\frac{1+x}{1-x}) = 2(x + \frac{x^3}{3} + \frac{x^5}{5} + ...) \quad (3)$$

from (3), $\ln x = 2[(\frac{x-1}{x+1}) + \frac{1}{3}(\frac{x-1}{x+1})^3 + \frac{1}{5}(\frac{x-1}{x+1})^5 + ...]$

In general, $\ln x = 2\sum_{k=1}^{\infty} \frac{1}{2k-1}(\frac{x-1}{x+1})^{2k-1}$

So, $log_b x$ an be implemented as $\frac{\ln x}{\ln b}$.

The Pseudo Code for Calculating Logarithm function using series expansion is shown below.

Algorithm 1 Pseudo Code for Calculating Logarithm Function

```
Require: value: $x > 0$ And base: $b \neq 1 \vee b > 0$          //where $x, b \in \mathscr{R}^+$
Ensure: $result = \log_b x$
function CalculatePower($base$, $exponent$)
    $power \leftarrow 1$
    for $i \leftarrow 1, exponent$ do
        $power \leftarrow power * base$
    end for
    return $power$          //It returns the base to the power exponent
end function

procedure CalculateNaturalLog($value$)
    $sum \leftarrow 0$
    $j \leftarrow (value - 1)/(value + 1)$
    for $i \leftarrow 1, \infty$ do
        $k \leftarrow (2 * i) - 1$
        $sum \leftarrow sum + (1/k) *$ CalculatePower($j, k$)
    end for
    return $2 * sum$          //It returns ln using series expansion.
end procedure

$a \leftarrow$ CalculateNaturalLog($x$){Calculates $lnx$}
$b \leftarrow$ CalculateNaturalLog($b$){Calculates $lnb$}
$result \leftarrow a/b$          //Final result of $log_b x$
=0
```

Advantages

- This form of series representation is Predictable and also it is easier to implement in programming aspect.
- This series is very useful for derivations

- It can be used to get theoretical error bounds.

Disadvantages

- Successive terms get very complex and hard to derive.
- Truncation error tends to grow rapidly away from expansion point

0.3.2 Algorithm 2

The another method of implementing logarithm function is using recursion which calculates log of value x to the base b.
The Pseudo Code for Calculating Logarithm function using simple recursion is shown below.

Algorithm 2 Pseudo Code for Calculating Logarithm Function

```
Require: value: $x > 0$ And base: $b \neq 1 \vee b > 0$ {where $x, b \in \mathscr{R}^+$}
Ensure: $result = \log_b x$                                   // Output will be integers.
function CalculateLog($value$, $base$)
    if $value > base - 1$ then
        return $1 +$ CalculateLog($value/base, base$)         //Recursive Call
    else
        return 0
    end if
end function

$result \leftarrow$ CalculateLog($x,, b$)                      // Final result of $log_b x$
=0
```

Advantages

- Complex code can be written easily in less code with Recursive method.

Disadvatages

- result is showing integer numbers only not any floating point.
- Recursive code is difficult to understand and debug.

- Terminating condition is must, otherwise it will go in infinite loop
- Execution speed decreases because of function call and return activity.

Bibliography

[1] https://en.wikipedia.org/wiki/Logarithm

[2] https://www.rapidtables.com/math/algebra/Logarithm.html