SOEN-6011 SOFTWARE ENGINEERING PROCESS

ETERNITY : FUNCTION : LOG$_B(X)$

Deliverable 1

SAHANA ANANTHA
Student Id : 40085533
Guided By - Prof. PANKAJ KAMTHAN
CONCORDIA UNIVERSITY
https://github.com/SahanaAnantha/SOEN-6011

Contents

1 Introduction

1.1 Description

A logarithmic answer the question " How many of this number do we multiply to get that number? "

For Example: How many 2s must we multiply to get 8?

$$Ans : 2 * 2 * 2$$

So we had to multiply 3 of the 2s to get 8
We say the logarithm of 8 with base 2 is 3
In fact, these two things are the same: 2 * 2 * 2 is equivalent to $log_2(8) = 3$

1.2 Definition

A logarithm is an exponent which indicates to what power a base must be raised to produce a given number. The logarithm of x in the base b is written $log_b(x)$ and is defined as,

$$log_b(x) = y$$

if and only if $by = x$, where $x > 0$ and $b > 0$, $b \neq 1$

logarithmic form : $log_b(x)$

exponential form : $b^y = x$

1.3 Domain

Set of positive real numbers $x > 0$

1.4 Co-Domain

Set of real numbers R from $-\infty$ to $+\infty$

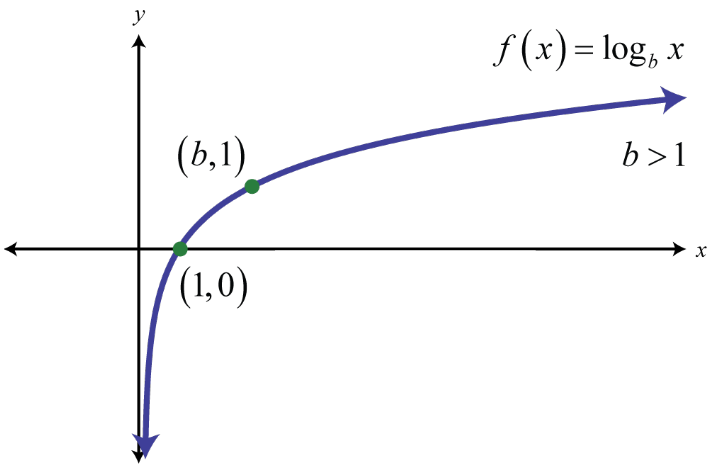

Figure 1: Graph of logarithmic function (Source: Google Images)

2 Functional Requirement

2.1 Assumption

- ID: 1
  - Description: The function $log_b(X)$ has two inputs that is b and X
- ID: 2
  - Description: Output of the function $log_b(X)$ is a Real Number
- ID: 3
  - Description: For out of range values of input , the output is undefined and throws error

2.2 Requirements

- ID: RQ1
  - Version: 1.0
  - Type: Functional Requirement
  - Priority: 1
  - Risk: High

  - Description: The user has to input the valid X and b values
  - Rationale: The input values provided by the user should be within the domain specified

- ID: RQ2
  - Version: 1.0
  - Type: Functional Requirement
  - Priority: 1
  - Risk: High
  - Description: When either the value of X or b are missing, the system should display error
  - Rationale: X and b are the mandatory input for log function

- ID: RQ3
  - Version: 1.0
  - Type: Functional Requirement
  - Priority: 1
  - Risk: High
  - Description: The value of b should be greater than 1.
  - Rationale: If the value of b is 0 the result would be undefined

- ID: RQ4
  - Version: 1.0
  - Type: Functional Requirement
  - Priority: 2
  - Risk: medium
  - Description: The console will display the result in stipulated time
  - Rationale: To measure the performance and efficiency.

- ID: RQ5
  - Version: 1.0
  - Type: Non- Functional Requirement
  - Priority: 3
  - Risk: Low
  - Description: The console should display appropriate error message when invalid input is given.
  - Rationale: To have better user friendly system

3 Algorithm

Algorithm 1 Approximation algorithm for LOGARITHM (X, b)

Input: X value $(X > 0)$ **; b value** $(b > 0, b \neq 1)$ $b = e = 2.1782$
Output: y has values between $-\infty$ **to** $+\infty$

procedure APPROXIMATION(X, e)
  if x is > 1.0 **then**
    $term \leftarrow (X-1)/X$
    $denominator \leftarrow (X-1)/X$
    $temp \leftarrow term$
    while $temp > 1E - 15$ **do**
      $result \leftarrow result + temp$
      $term \leftarrow term * (1.0/denominator)$
      $denominator \leftarrow denominator + 1$
    return $result$
  if x is greater 0.0 **then**
    $result \leftarrow 0.0$
    $term \leftarrow X - 1$
    $Powerofone \leftarrow -1$
    $denominator \leftarrow 2$
    $temp \leftarrow term$
    while $temp > 1E - 15|| - temp > 1E - 15$ **do**
      if $temp > 1E - 15$ **then**
        $result \leftarrow result - temp$
  Else
  $result \leftarrow result + temp$
  $term \leftarrow term * (X - 1)$
  $temp \leftarrow term * powerof1$
  $denominator \leftarrow denominator + 1$
  return $result + term$
  return $result$

Algorithm 2 Algorithm for LOGARITHM (X)

Input: X value $(X > 0)$ **; b value** $(b > 0, b \neq 1)$ $b = e = 2.1782$
Output: LnofX has values between $-\infty to + \infty$

```
procedure APPROXIMATION()X, e
    X ← Input                                   ▷ input the value of X
    y ← (X−1)/(X+1)
    ySquared ← y ∗ y
    increment ← 1
    for (i=1/increment i> .00000000001; increment = increment+2 ) do
        repeatingValue ← i + ySquared
        newRepeat ← repeatingValue
        finalRepeat ← newRepeat ∗ repeatingValue
        LnofX = 2 ∗ y ∗ finalRepeat
```

3.1 Description

For calculating the Logarithmic value , the above two algorithms are selected.

3.2 Algorithm 1

The first one chooses the approximation algorithm which uses the constant variables as input. In this the value of e that is the Euclid's constant is declared and accordingly the pseudo code tells the way in which the calculation of natural logarithm is made. Similarly the base value is calculated as below.

$$log_b(x) = \frac{log_e(x)}{log_e(b)}$$

3.3 Algorithm 2

The second algorithm for the Logarithmic function has the steps to calculate the value of log with base e. With the help of these steps the logarithm value to different base values can be calculated.The result calculated will be in decimals.

$$log_b(x) = \frac{log_e(x)}{log_e(b)}$$

3.4 Advantages and Disadvantages

Advantages

- Algorithm1 provides the more accurate results for the value of x with in the domain
- The values can be separated and calculated for each input using the natural logarithm
- Execution is faster in algorithm1 than algorithm2 which can be estimated from the O notation.

Disadvantages

- The Algorithm2 uses the for loop with with less precision values which hinders the performance

Bibliography

[1] University College Cork,
http://www.cs.ucc.ie/d̃gb/courses/toc/handout26.pdf

[2] Basecs,
https://medium.com/basecs/looking-for-the-logic-behind-logarithms-9e79d7666dda

[3] Wikipedia,
https://en.wikipedia.org/wiki/Natural_logarithm#Numerical_value

[4] University of Sydney,
https://magma.maths.usyd.edu.au/magma/handbook/text/194