

F4: Logarithmic Function

MANIKANDAN SHANMUGAM

40070772

Problem 1

SOEN 6011

Deliverable 1

Introduction

The *exponential function*: $y = b^x$ is a one-to-one function, which means that for each x there is only one y and for each y there is only one x . Functions that are one-to-one have *inverse functions*. *Logarithmic functions* are the inverses of exponential functions.

Definition

To express 'y' as a function of 'x' the logarithm was invented. The formal definition of a **logarithmic function** is,

$$\log_b(x) = y \quad (1)$$

The base 'a' logarithm of a positive number 'x' is the exponent you get when you write 'x' as a power of 'a'. This is equivalent to saying "y is the base-b logarithm of x."

Domain and Co-Domain

The domain of the logarithmic function $y = \log_b(x)$ is the set of positive real numbers $(0, \infty)$ with domain restrictions of $x > 0$ and $x \neq 1$.

The co-domain is the set of real numbers $(-\infty, \infty)$.

Characteristics

- The function will always have an x-intercept of one, occurring at the ordered pair of $(1, 0)$.
- In a logarithmic graph, the "rate of change" increases or decreases across the graph.
- There is no y-intercept with the function since it is asymptotic to the y-axis.
- When $b > 1$, the graph increases and when $0 < b < 1$, the graph decreases.

Problem 2

Assumptions

- In function $\log_b(x) = y$, b is a positive real number greater than 0.
- In function $\log_b(x) = y$, x is a positive real number greater than 0 and $\neq 1$.
- Function returns a real number value ranging $(-\infty, \infty)$.
- If the inputs are not within the assumption constraints, the function does not give the desired output.

Requirements

1. First Requirement

- ID : FR1
- TYPE : Functional Requirement
- VERSION : 1.0
- DIFFICULTY : Easy
- DESCRIPTION : System shall take an input value for b as a base value for the function $\log_b(x)$.
- RATIONAL : The function $\log_b(x)$ can not be defined if the base value is not a positive real number inclusive of 0.

2. Second Requirement

- ID : FR2
- TYPE : Functional Requirement
- VERSION : 1.0
- DIFFICULTY : Easy
- DESCRIPTION : System shall take an input value for x for the function $\log_b(x)$.
- RATIONAL : The function $\log_b(x)$ can not be defined if the value of x is not a positive real number inclusive of 0 exclusive of 1.

3. Third Requirement

- ID : FR3
- TYPE : Functional Requirement
- VERSION : 1.0
- DIFFICULTY : Nominal
- DESCRIPTION : System shall calculate the output value for the function $\log_b(x)$ and limit it to two digit values after decimal.
- RATIONAL : The function $\log_b(x)$ must give desired output and might give an output with endless decimal points.

4. Fourth Requirement

- ID : FR4
- TYPE : Functional Requirement
- VERSION : 1.0
- DIFFICULTY : Nominal
- DESCRIPTION : System shall not accept an invalid input value other than the input boundaries given in assumption for the function $\log_b(x)$ and throw an error message.
- RATIONAL : The function $\log_b(x)$ can not be calculated with inputs outside of the assumptions and invalid inputs (string, character, etc.).

Problem 3

Pseudo code and Algorithm

Calculate: $\log_b(x)$

Algorithm 1 Calculate Log function using Recursion

```
1. function logCalculator(base,x)
in:  int number base, double number x
out: double number output
2. if  $base > 0$  then
3.     if  $base > 0$  then
4.         return 0
5.     else
6.         return  $1 + \logCalculator(base, \text{int}(x/base))$ 
7.     end
8. else
9.     return 0
10.end
```

Algorithm 2 Calculate Log function using Loop

```
1. function logCalculator(x)
in:  int double number x
out: double number output
2.  $temp \leftarrow (x - 1)/(x + 3)$ 
3.  $sum \leftarrow 1$ 
4.  $temppow \leftarrow temp$ 
5. for  $n \leftarrow 1, 100$  do
6.     if  $n \bmod 2 \neq 0$ 
7.          $sum \leftarrow sum + temppow/n$ 
8.     end
9.      $temppow \leftarrow temppow * temp$ 
10.end for
11.  $sum \leftarrow sum * 2$ 
12. return  $sum$ 
13.end
```

Advantages and Disadvantages

Algorithm 1:

Advantages:

- Recursion makes code smaller.
- Recursion is easy to understand, and it has excellent performance on readability.
- In code, the purpose of recursion is clear than loop.
- Recursion has higher maintainability than loop.

Disadvantages:

- Recursion repeatedly invokes the mechanism, and consequently the overhead, of method calls. This can be expensive in both processor time and memory space.
- Recursion could lead the problem of memory overflow.
- Infinite recursion can lead to system crash.
- Recursion needs system continuously allocates memory space, thus it has a bad effect on efficiency.

Algorithm 2:

Advantages:

- Loop is less expensive to maintain than recursion.
- Loop avoids of memory overflow.
- Loop could avoid memory overflow of input. Thus the value of input is unrestricted.
- Loop needs less time to execute. Besides, it also needs less memory.

Disadvantages:

- It uses more parameters, and the structure is more complex than recursion.
- In readability, loop is weak, because of its complex code structure.
- Infinite loop iteration consumes CPU cycles.

Conclusion

Recursion is generally used because of the fact that it is simpler to implement, and it is usually more elegant than iterative solutions. Remember that anything that's done in recursion can also be done iteratively, but with recursion there is generally a performance drawback. But, depending on the problem that you are trying to solve, that performance drawback can be very insignificant in which case it makes sense to use recursion. With recursion, you also get the added benefit that other programmers can more easily understand your code which is always a good thing to have. As a result, the recursion is better.

Pseudo code for Algorithm 1

This program calculates the Logarithmic values for double input value

```
function logCalculate(Argument one, Argument two)
```

Calculate the log value of Argument 1(base) and Argument 2(x) by calling the log-Calculate function recursively by passing parameters logCalculate(base, x/base)

```
end
```

In the main function

```
print prompt "Input two numbers"
```

Take the first number from the user
Take the second number from the user

Send the first number and second number to the logCalculate function and print the result to the user

References

- [1] Hotmath,
https://www.varsitytutors.com/hotmath/hotmath_help/topics/domain_and_range_of_exponential_and_logarithmic_functions
Logarithmic functions,
<http://www.biology.arizona.edu/biomath/tutorials/Log/Definition.html>
- [2] Tutorialspoint,
https://www.tutorialspoint.com/java/lang/math_pow.htm
Logarithmic functions,
<http://www.biology.arizona.edu/biomath/tutorials/Log/Definition.html>
- [3] Chem869Mats,
<http://dwb4.unl.edu/Chem/CHEM869R/CHEM869RMats/Logs/Logs.html>