Data Structure and Algorithm

Laboratory Activity No. 2

Algorithm Analysis and Flowchart

Submitted by:
Barbas, Steven Jade P.

Instructor:
Engr. Maria Rizette H. Sayo

July, 30, 2025

I. Objectives

Introduction

       Data structure is a systematic way of organizing and accessing data, and an algorithm is a step-by-step procedure for performing some task in a finite amount of time. These concepts are central to computing, but to be able to classify some data structures and algorithms as "good," we must have precise ways of analyzing them.

This laboratory activity aims to implement the principles and techniques in:
- Writing a well-structured procedure in programming
- Writing algorithm that best suits to solve computing problems to improve the efficiency of computers
- Convert algorithms into flowcharting symbols

II. Methods

A. Explain algorithm and flowchart

B. Write algorithm to find the result of equation: $f(x) = \begin{cases} -x, & x<0 \\ x, & x \geq 0 \end{cases}$ and draw its flowchart

C. Write a short recursive Python function that finds the minimum and maximum values in a sequence without using any loops

III. Results

A.

       An algorithm is a step-by-step set of instructions used to solve a problem or perform a task, like finding a number in a list using linear search by checking each element one at a time. A flowchart, on the other hand, is a visual representation of an algorithm using shapes and arrows to show the flow of the process. While algorithms are written in plain text and can be a bit harder to understand, flowcharts make it easier to visualize the logic of a program. Both are important tools in programming because they help in planning and understanding how a program works.

B. **Algorithm**

- Start
- Input value of x
- If x < 0, then f(x) = -x
- Else, f(x) = x

1

- Output f(x)

- End

Flowchart:
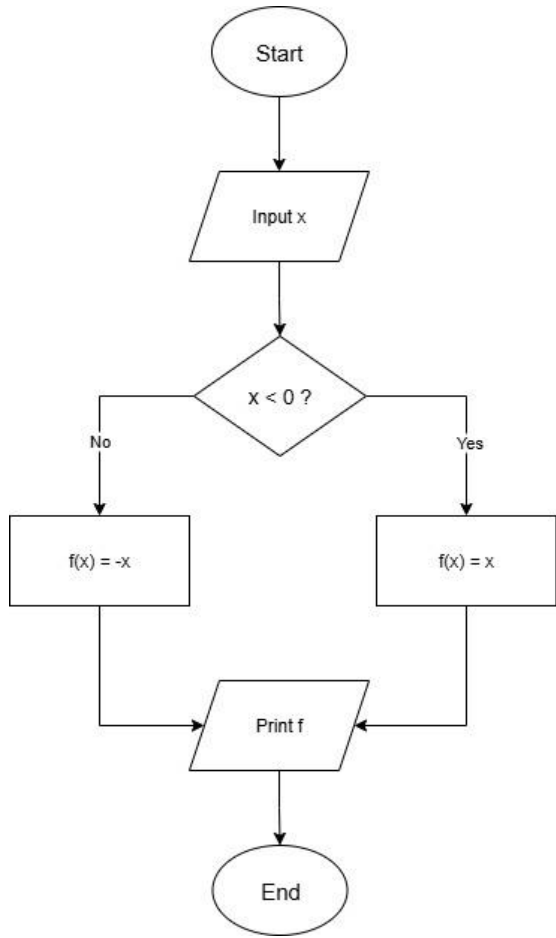
C.   Program

```python
def minmax(arr):
    if len(arr) == 1:
        return arr[0], arr[0]

    a, b = minmax(arr[1:])

    return min(arr[0], a), max(arr[0], b)

nums = [5, 2, 9, 1, 7]
x, y = minmax(nums)
print(f"Min: {x}, Max: {y}")

z, w = minmax([42])
print(f"Min: {z}, Max: {w}")
```

```
Min: 1, Max: 9
Min: 42, Max: 42
```

Figure 1 Screenshot of program

Program Link: LAB-2-C - Colab

IV. Conclusion

In conclusion, this activity helped me gain a better understanding of how algorithms and flowcharts work hand in hand to solve problems. Writing a basic algorithm and creating its corresponding flowchart allowed me to clearly see how each step is carried out. I also learned how to apply recursion in Python to find the minimum and maximum values in a list without using loops, demonstrating the power and versatility of programming.

References

[1] Co Arthur O.. "University of Caloocan City Computer Engineering Department Honor Code," UCC-CpE Departmental Policies, 2020.