UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 2

Algorithm Analysis and Flowchart

Submitted by:
Talagtag Mark Angel T.

Instructor:
Engr. Maria Rizette H. Sayo

AUGUST 2, 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

Algorithm and Flowchart Algorithm An algorithm is a step-by-step set of instructions for solving a specific problem or performing a task. Written in plain language or pseudocode Must be finite and precise Should have a clear start and end Flowchart A flowchart is a diagram that visually represents an algorithm using symbols: Oval → Start/End Parallelogram → Input/Output Rectangle → Process/Action Diamond → Decision/Condition

Algorithm Steps

Start

Input a number x

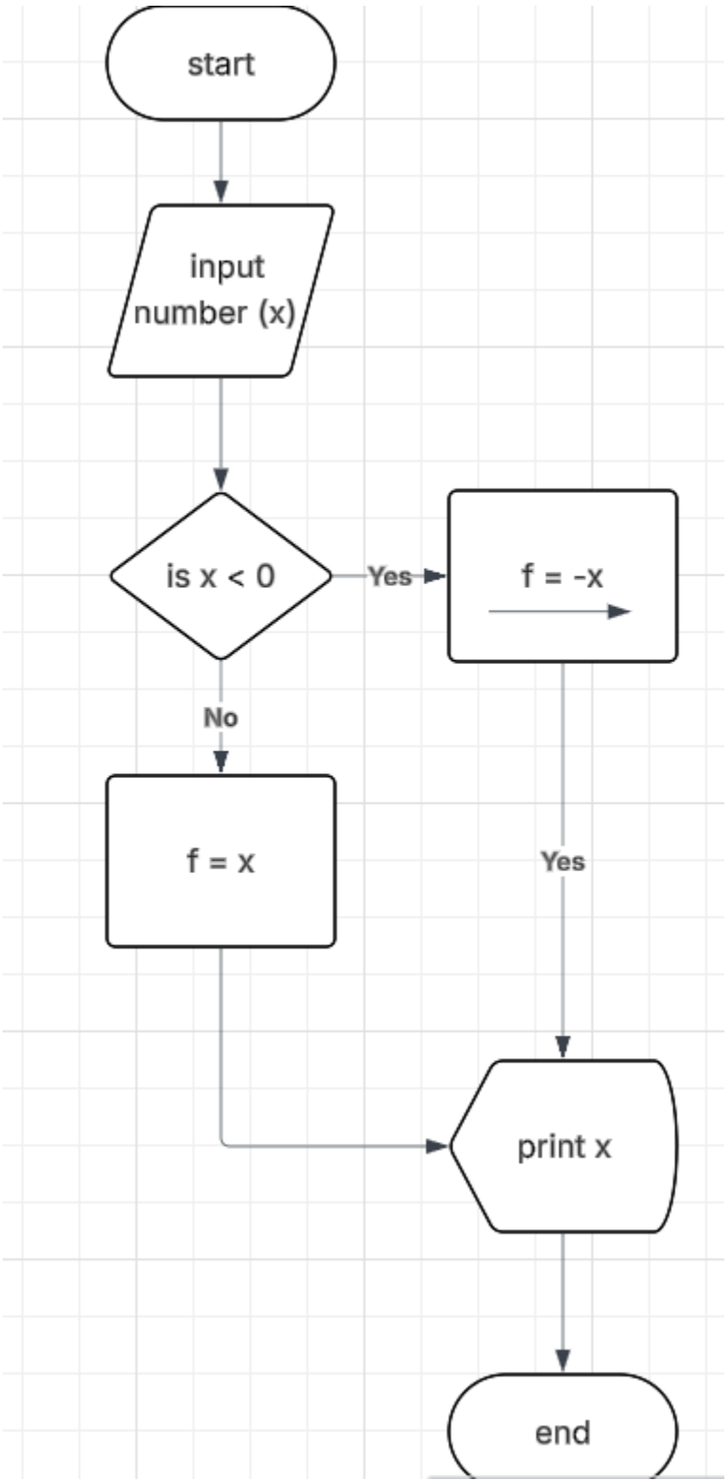
If x < 0, then set f = -x

Else, set f = x

Output f

End

Flow Chart



```python
def find_min_max(seq):
    length = len(seq)

    if length == 1:
        single_value = seq[0]
        return single_value, single_value
    else:
        first_value = seq[0]
        remaining_values = seq[1:]

        min_rest, max_rest = find_min_max(remaining_values)
        min_value = min(first_value, min_rest)
        max_value = max(first_value, max_rest)

        return min_value, max_value


# Example usage
numbers = [3, 1, 9, -5, 7]
minimum, maximum = find_min_max(numbers)
print("Minimum:", minimum)
print("Maximum:", maximum)
```

```
Minimum: -5
Maximum: 9
```

Figure 1 Screenshot of program

IV. Conclusion

I learned the fundamental concepts of **algorithms** and **flowcharts**, both of which are essential tools in problem-solving and computer programming. An algorithm is a step-by-step procedure or a set of rules designed to solve a problem or perform a task. It serves as the blueprint for writing computer programs. A flowchart, on the other hand, visually represents an algorithm. It uses different shapes like rectangles, diamonds, and arrows to describe the flow of operations, making it easier to understand the steps involved. We also explored the process of creating an algorithm to find the result of an equation. The algorithm defines a clear sequence of operations to calculate the desired output. We followed this with a flowchart that visually represents the steps of this algorithm, helping to simplify and communicate the process.

In summary, algorithms and flowcharts are essential in planning and organizing a problem-solving approach, while recursion in Python offers an efficient way to handle complex tasks with minimal code. These concepts and techniques form the foundation for more advanced programming and computational problem-solving.

References

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