



UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT



Data Structure and Algorithm

Laboratory Activity No. 2

Algorithm Analysis and Flowchart

Submitted by:
Asugas, Kenneth R.

Instructor:
Engr. Maria Rizette H. Sayo

07, 28, 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

- Explain algorithm and flowchart
- 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
- Write a short recursive Python function that finds the minimum and maximum values in a sequence without using any loops

III. Results

Present the visualized procedures done. Also present the results with corresponding data visualizations such as graphs, charts, tables, or image . Please provide insights, commentaries, or explanations regarding the data. If an explanation requires the support of literature such as academic journals, books, magazines, reports, or web articles please cite and reference them using the IEEE format.

Please take note of the styles on the style ribbon as these would serve as the style format of this laboratory report. The body style is Times New Roman size 12, line spacing: 1.5. Body text should be in Justified alignment, while captions should be center-aligned. Images should be readable and include captions. Please refer to the sample below:

Algorithm:

Step 1: Start

Step 2: Input x

Step 3: If $x < 0$, then $\text{result} = -x$

Step 4: Else, $\text{result} = x$

Step 5: Output result

Step 6: End

The algorithm computes the absolute value of a given number x . It checks if $x < 0$, if true it multiplies by -1 to make it positive. Otherwise, it returns x as is.

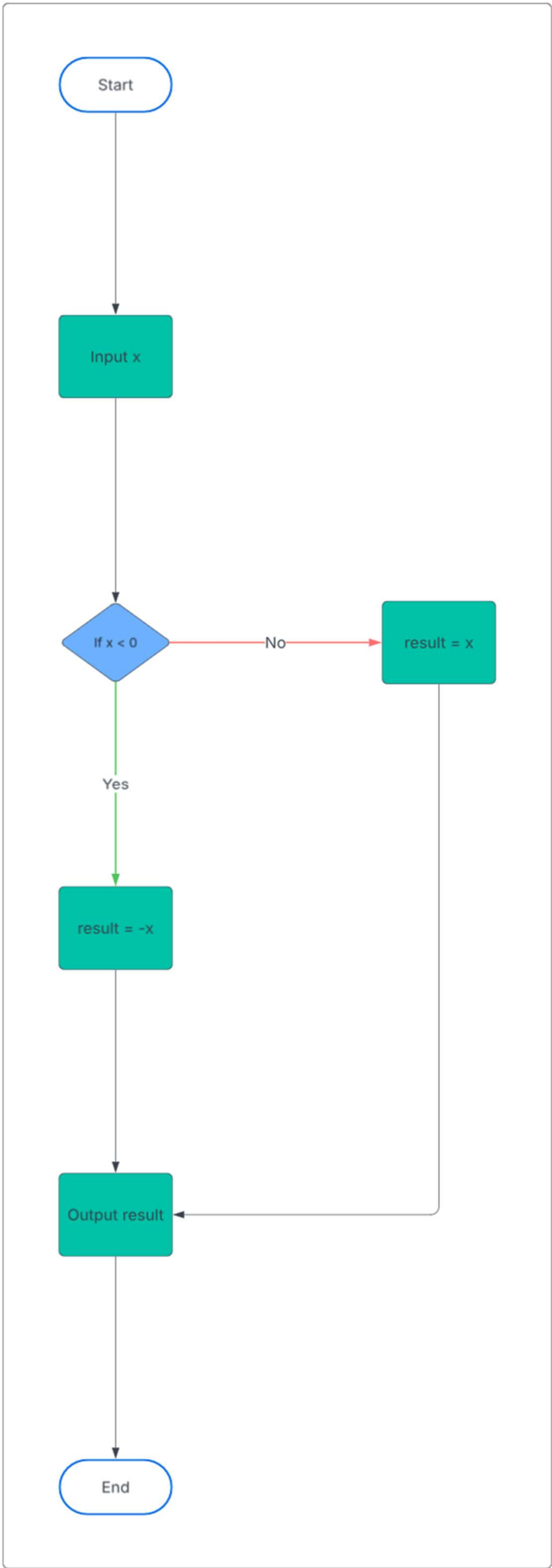
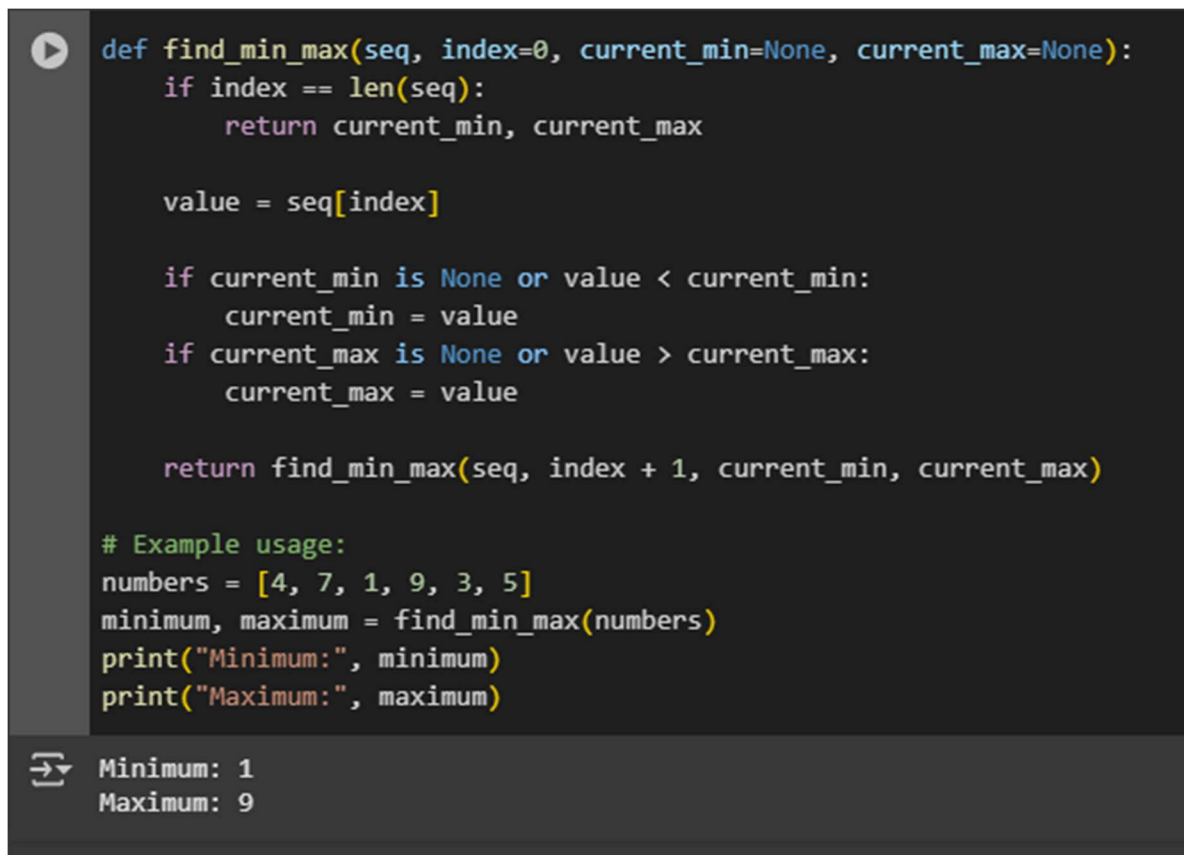


Figure 1 Screenshot of flowchart

The flowchart starts with inputting x , proceeds to a decision block checking if $x < 0$. Then follows either the “Yes” path (set result = $-x$) or “No” path (set result = x). Outputs the result and ends.



```
def find_min_max(seq, index=0, current_min=None, current_max=None):
    if index == len(seq):
        return current_min, current_max

    value = seq[index]

    if current_min is None or value < current_min:
        current_min = value
    if current_max is None or value > current_max:
        current_max = value

    return find_min_max(seq, index + 1, current_min, current_max)

# Example usage:
numbers = [4, 7, 1, 9, 3, 5]
minimum, maximum = find_min_max(numbers)
print("Minimum:", minimum)
print("Maximum:", maximum)
```

Minimum: 1
Maximum: 9

Figure 2 Screenshot of program

This recursive function scans the list element by element without loops, updating the smallest and largest values until the sequence ends.

IV. Conclusion

The laboratory exercise reinforced the fundamentals of algorithm creation, flowchart representation, and Python programming using recursion. Implementing $f(x)$ clarified the concept of conditional branching, while the recursive min-max function highlighted how problem-solving can be achieved without traditional loops. This activity strengthened logical thinking, coding discipline, and the ability to translate problem statements into clear procedural steps.

References

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