Data Structure and Algorithm

Laboratory Activity No. 2

Algorithm Analysis and Flowchart

Submitted by:
Palmes, Lewis Clark L.

Instructor:
Engr. Maria Rizette H. Sayo

July 26, 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.   Explain algorithm and flowchart

Algorithm

       An algorithm is a finite set of step-by-step instructions written to perform a specific task or solve a problem. It must be clear, unambiguous, and have a definite end.

Flowchart

       A flowchart is a diagrammatic representation of an algorithm. It uses standardized symbols like ovals (start/end), rectangles (process), diamonds (decision), and arrows (flow) to show the sequence of steps.

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

Algorithm

1.   Start
2.   Input a number f(x)
3.   If f(x) < 0, then f(x) = -x
4.   Else, set f(x) = x
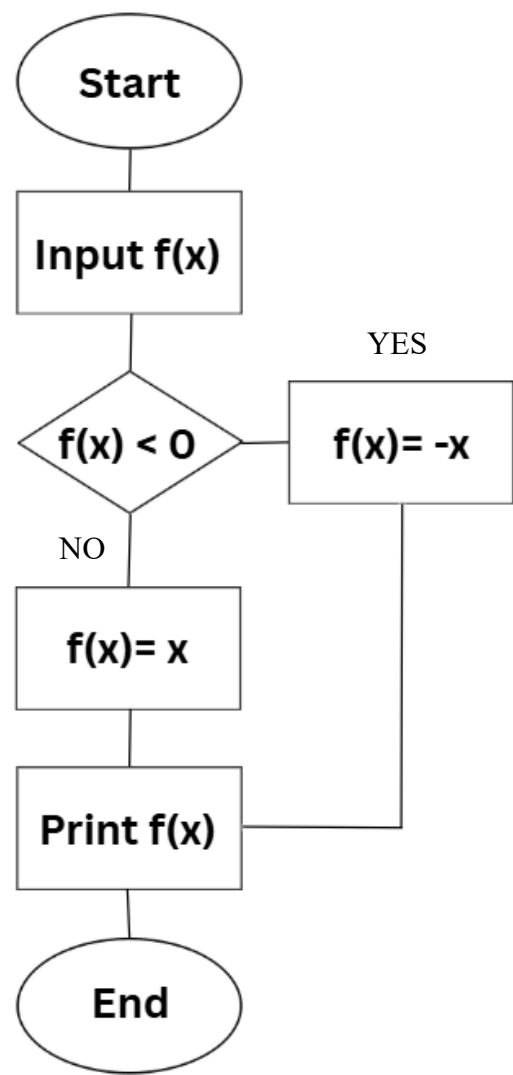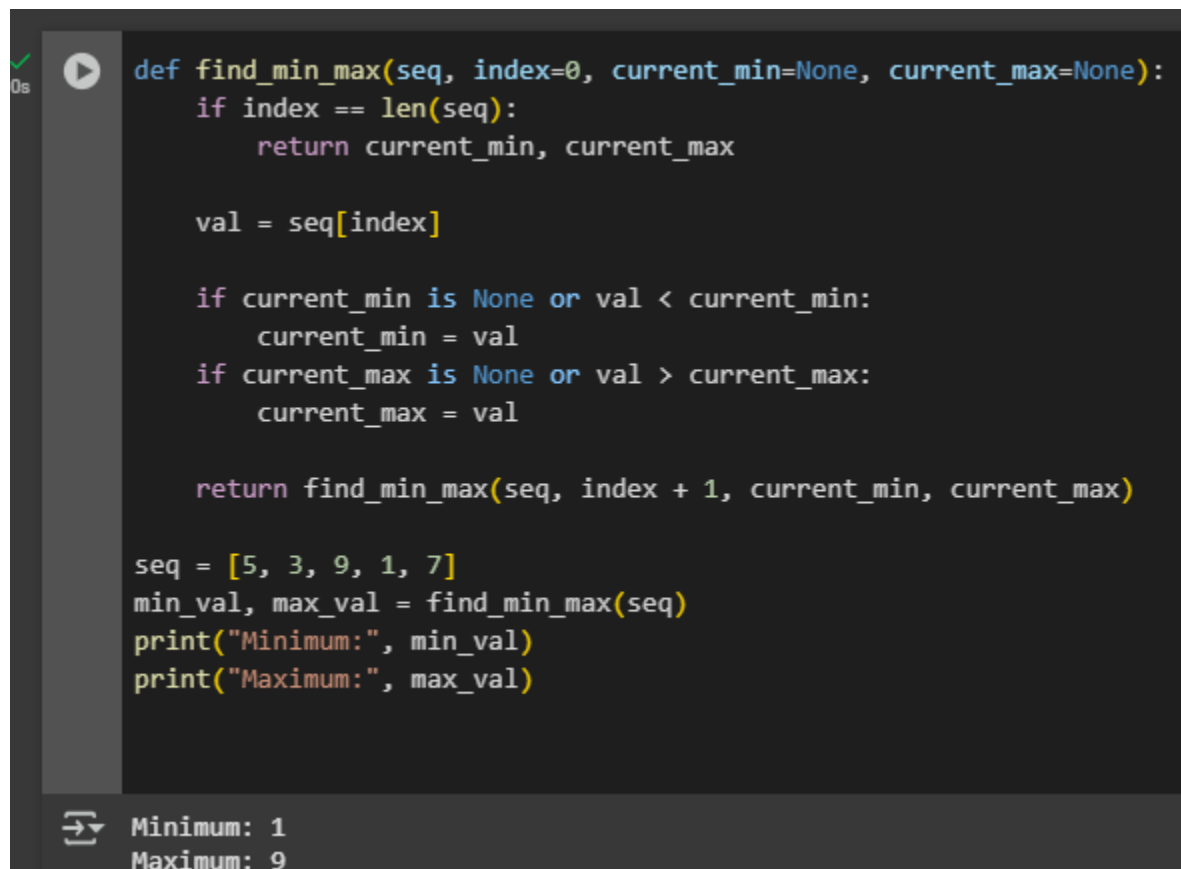5.   Print f(x)
6.   End

Flowchart

Figure 1 Flowchart (Canva)

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

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

    val = seq[index]

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

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

seq = [5, 3, 9, 1, 7]
min_val, max_val = find_min_max(seq)
print("Minimum:", min_val)
print("Maximum:", max_val)
```

```
Minimum: 1
Maximum: 9
```

Figure 2 Screenshot of program

The image displays a Python script that defines a recursive function named find_min_max, which determines the smallest and largest values in a list of numbers. It uses optional parameters to track the current index, minimum, and maximum values while iterating through the list. A sample list [5, 3, 9, 1, 7] is passed to this function, and it prints the result: the minimum is 1 and the maximum is 9.

IV. Conclusion

This laboratory activity provided a deeper understanding of fundamental concepts in data structures and algorithms, particularly focusing on writing clear and efficient algorithms and representing them visually through flowcharts. By implementing a recursive function to find the minimum and maximum values in a sequence, we demonstrated the power of recursion in solving problems without using loops. The ability to break down problems into structured, logical steps and represent them visually enhances the clarity and efficiency of computational solutions.

References

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

[2] Google Colab. (n.d.). https://colab.research.google.com/drive/1hv7I4T8CyZFzJZXU36LqdJAXKp0g7Jjy#scrollTo=wKTdapKeDSaI