



UNIVERSITY OF CALOOCAN CITY  
COMPUTER ENGINEERING DEPARTMENT



Data Structure and Algorithm

Laboratory Activity No. 2

---

# Algorithm Analysis and Flowchart

---

*Submitted by:*  
Disomnong, Jalilah M.

*Instructor:*  
Engr. Maria Rizette H. Sayo

07-26-2025

# 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.

An algorithm is a collection of limited guidelines or directives that must be adhered to when doing computations or other problem-solving procedures. Additionally, it is a process that often uses recursive operations to solve a mathematical problem in a limited number of steps (GeeksforGeeks, 2025c).

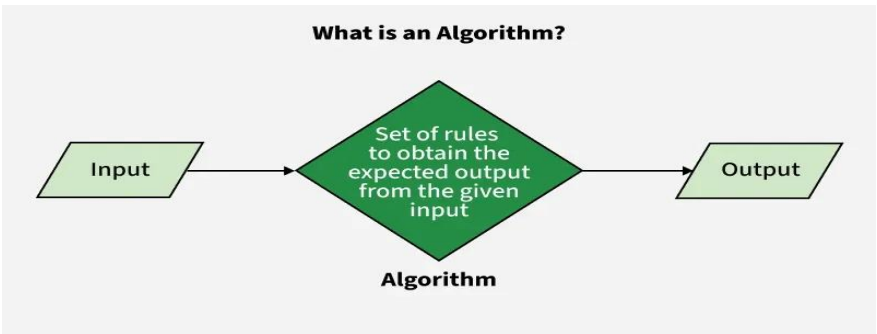


Figure 1 [https://media.geeksforgeeks.org/wp-content/uploads/20250429120929479720/What-is-an-Algorithm\\_.webp](https://media.geeksforgeeks.org/wp-content/uploads/20250429120929479720/What-is-an-Algorithm_.webp)

Flowcharts are visual tools that represent data, algorithms, or processes in a structured and easy-to-understand manner. They break down complex problems into step-by-step solutions, making them particularly useful for beginner programmers. Additionally, flowcharts are helpful in debugging and troubleshooting by clearly outlining the flow of a program or system. Typically, they consist of sequentially arranged boxes that illustrate the progression of a process. Because they visually convey an algorithm or workflow, flowcharts are easier to interpret than text-based descriptions. However, to ensure they are effective and universally understood, it is important to follow standardized rules that promote clarity and consistency across various users and industries (GeeksforGeeks, 2025a).

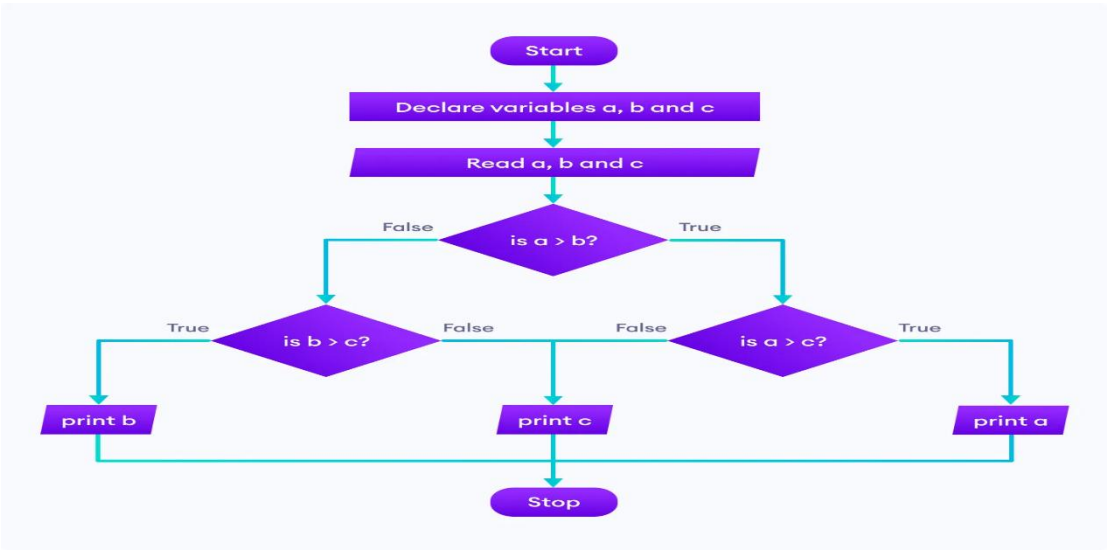


Figure 2 Example <https://cdn.programiz.com/sites/tutorial2program/files/flowchart-2.jpg>

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

Step 1: Start

Step 2: Input Value of X

Step 3: If  $x < 0$  then

Set  $f(x) = x$

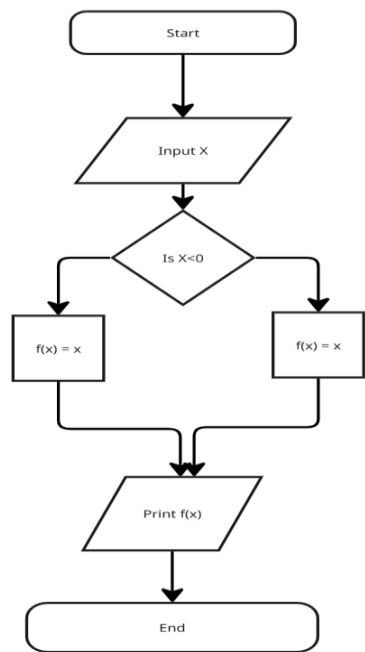
Step 4: Else if  $x \geq 0$

Set  $f(x) = x$

Step 5: Output value of function

Step 6: End

Flowchart



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

```
class Number:
    def __init__(self):
        self.num = []

    def my_list(self):
        self.num = [54, 34, 76, 87, 64, 45]
        return self.num

def find_min_max(seq):
    match seq:
        case [single]:
            return single, single
        case [first, *rest]:
            min_rest, max_rest = find_min_max(rest)
            return min(first, min_rest), max(first, max_rest)

list = Number()
list.my_list()

min_val, max_val = find_min_max(list.num)
print("List:", list.num)
print("Minimum:", min_val)
print("Maximum:", max_val)
```

List: [54, 34, 76, 87, 64, 45]  
Minimum: 34  
Maximum: 87

Figure 3 <https://colab.research.google.com/drive/1Kxi2JcavWU5z24TslrOG8oHK1fY9dyIs>

The code defines a class `Number` that initializes an empty list and fills it with a fixed set of integers using the `my_list` method. The `find_min_max` function uses Python's match-case pattern matching to recursively find the minimum and maximum values in the list. When the list has only one element, it returns that element as both the minimum and maximum. If the list has more elements, it splits the list into the first element and the rest, then recursively finds the min and max of the rest and compares them with the first element to update the results.

## IV. Conclusion

In this laboratory, we learned the fundamental concepts of algorithm design and flowchart creation. Specifically, we wrote an algorithm and drew its corresponding flowchart, and implemented a recursive Python function to find the minimum and maximum values in a list without using loops or conditional statements. These exercises helped us better understand how to translate logic into algorithms and visualize processes using flowcharts.

## References

- [1] GeeksforGeeks, “What is an Algorithm | Introduction to Algorithms,” *GeeksforGeeks*, Jul. 11, 2025.  
<https://www.geeksforgeeks.org/dsa/introduction-to-algorithms/>
- [2] GeeksforGeeks, “What is a Flowchart and its Types?,” *GeeksforGeeks*, Apr. 07, 2025.  
<https://www.geeksforgeeks.org/computer-science-fundamentals/what-is-a-flowchart-and-its-types/>