Adamson University
College of Engineering
Computer Engineering Department

Data Structure and Algorithm

Laboratory Activity No. 2

Algorithm Analysis and Flowchart

Submitted by:
Manigbas, Jeus Miguel T. <Leader>
Barredo, Alwin P.
Bela, Lorenzo Miguel D.
Callorina, Robert Victor A.
Guzon, Kean Louiz I.
.

Instructor:
Engr. Maria Rizette H. Sayo

November 08, 2023

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 set of well-defined instructions or steps that describe how to perform a task or solve a problem. It is a clear, concise, and unambiguous set of instructions that can be followed by anyone to achieve the desired outcome. Algorithms are essential for programming and problem-solving, as they provide a clear and structured way to break down a complex task into smaller, more manageable steps. While a flowchart is a graphical representation of an algorithm. It uses shapes and symbols to represent different steps and decision points in the algorithm. Flowcharts are easier to understand and follow than algorithms written in text form, as they provide a visual representation of the process. This can be especially helpful for complex algorithms or algorithms that involve decision-making.

B. Algorithm:

1. Start

2. Read the value of x.
3. If x is less than 0, set f(x) to -x.
4. If x is greater than or equal to 0, set f(x) to x.
5. Display the value of f(x).
6. End

Flowchart:

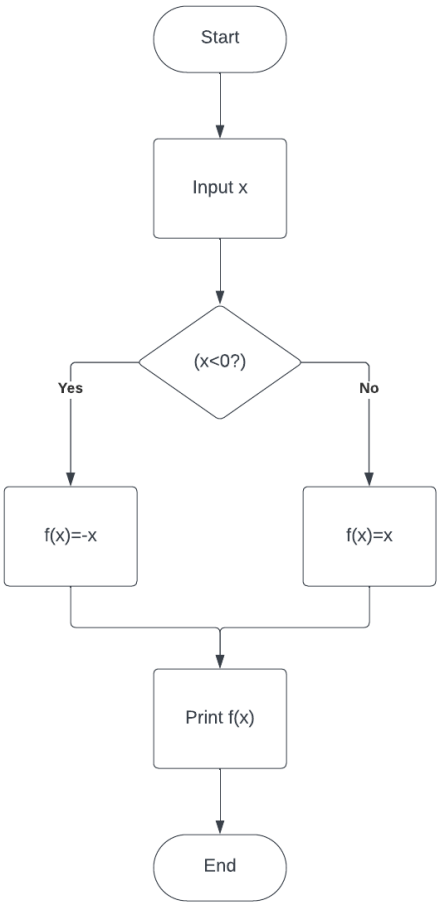

Figure 1. Screenshot of the Flowchart

C.

```
def minmax(a, b):
  if a <= b:
    return a, b
  else:
    return b, a

def minmaxseq(arr, n):
  if n == 1:
    return arr[0], arr[0]
  a, b = minmaxseq(arr, n-1)
  return minmax(a, arr[n-1])

arr = [10, 40, 20, 50, 30]
n = len(arr)
print("The minimum and maximum values are", minmaxseq(arr, n))
```

```
The minimum and maximum values are (10, 30)
```

Figure 2. Screenshot of the recursive Python function

IV. Conclusion

In this lab report, we dealt with algorithms, flowcharts, and presented a small recursive python program to find the minimum and maximum values in a sequence without using any loops. In conclusion, this lab report covers the principles of algorithms and flowcharts a discussion of importance, as well as examples of how to do both. It also shows how to use repetition to easily determine the minimum and maximum of a sequence in Python, without the need for loops. This method can be useful in applications where it is important to identify repeating elements in the sequence.

References

[1] Gillis, A.S. (2023) *What is an algorithm?: TechTarget*, *WhatIs.com*. Available at: https://www.techtarget.com/whatis/definition/algorithm (Accessed: 08 November 2023).

[2] *What is a flowchart? process flow diagrams & maps | ASQ*. Available at: https://asq.org/quality-resources/flowchart (Accessed: 08 November 2023).