

**Module Code & Module Title**

**CC4002NT, INFORMATION SYSTEM**

**Assessment Weightage &Type**

**20% and Individual**

**Year and Semester**

**2019 Spring**

**Student Name: Girija Tamang**

**London Met ID:18030995**

**Assignment Due Date: week 5**

**Assignment Submission Date: week 7**

**Word Count (Where Required):**

*I confirm that I understand my coursework needs to be submitted online via Local Server under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.*

Contents

[1. The Bubble Sort Algorithm 1](#_Toc5365341)

[1.1 Introduction 1](#_Toc5365342)

[1.2 Algorithm 2](#_Toc5365343)

[2.Graphic representation of the algorithm 3](#_Toc5365344)

[2.1Flowchart: 3](#_Toc5365345)

[3.Data Structures 4](#_Toc5365346)

[3.1 Primitive Data Structures 4](#_Toc5365347)

[3.1.1Integers 4](#_Toc5365348)

[3.1.2 Float 4](#_Toc5365349)

[3.1.3 String 4](#_Toc5365350)

[3.1.4 Boolean 5](#_Toc5365351)

[3.2 Non-Primitive Data Structures 5](#_Toc5365352)

[3.2.1 List 5](#_Toc5365353)

[3.2.2 Tuple 5](#_Toc5365354)

[3.2.3. Set 6](#_Toc5365355)

[3.2.4. Dictionary 6](#_Toc5365356)

[3.2.5. Array 6](#_Toc5365357)

[4.Learning Reflection 8](#_Toc5365358)

[5. References 9](#_Toc5365359)

**Figure**

[Figure 1: flowchart 3](#_Toc5365360)

# The Bubble Sort Algorithm

## 1.1 Introduction

Bubble Sort is easiest sorting algorithm that is primarily based on the concept of again and again comparing pairs of adjacent elements of an array and then swapping their positions if they exist in the incorrect order. We repeat this process till the array is sorted in ascending or descending order in python or other programming languages (Gupta, 2017).

Let the array be [9, 2, 7, 4, 6]. We know that 9 must not be on the left of 2 and so, we swap them and get [2, 9, 7, 4, 6].Next, we see that 9 need to once more no longer be on the left of 7 so we swap 7 and 9 then we get [2, 7, 9, 4, 6].we repeat this for 9 and 4 and as a result for 9 and 6 to get [2 7,4,6,9].

As can be seen – after one “pass” over the array, the greatest factor (9 in this case) has reached its correct position – extreme right. Let us strive to repeat this process. (2,7) is in right order. However (7,4) is a wrong order. Therefore, we swap 7 and 4 to get [2,4,7,6,9]. Now again (7,6) is improper so we do swap again and get [2,4,6,7,9]. As we see the array is sorted. In this way bubble sort exactly works (Sehgal, 2018). A figure is well worth a thousand words so, renowned this figure for higher understanding of working of bubble sort.

First Pass

Swapping

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 9 | 2 | 7 | 4 | 6 |

Swapping

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 9 | 7 | 4 | 6 |

Swapping

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 7 | 9 | 4 | 6 |

Swapping

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 7 | 4 | 9 | 6 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 7 | 4 | 6 | 9 |

Second Pass

No Swap

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 7 | 4 | 6 | 9 |

Swapping

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 7 | 4 | 6 | 9 |

Swapping

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 4 | 7 | 6 | 9 |

No Swap

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 4 | 6 | 7 | 9 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 4 | 6 | 7 | 9 |

The array is sorted.

## 1.2 Algorithm

The way to resolve a problem step by step is algorithm which is commonly is to calculate, process and other related computer operations. The algorithm of bubble sort for sorting array or list in ascending order is:

Step 1: Start

Step 2: Take the list of number from user that are to be sorted.

Step 3: After getting the number from user store all numbers in array named g.

Step 4: Creating a key that start with the second element of array g.

Step 5: Now comparing the (key­-1) element of g array with the (key) element of an array g.

Step 6: If the (key-1) element of the g array is larger than the (key) element of the g array, swap the respective elements of the g array.

Step 7: Now move the key element to the right to next element of an array g.

Step 8: Repeat the step 5 until the g array element previously sorted.

Step 9: Repeat the step 4 till (entered numbers -1) times.

Step 10: All the numbers that are entered in list are now sorted.

Step 11: Stop.

# 2.Graphic representation of the algorithm

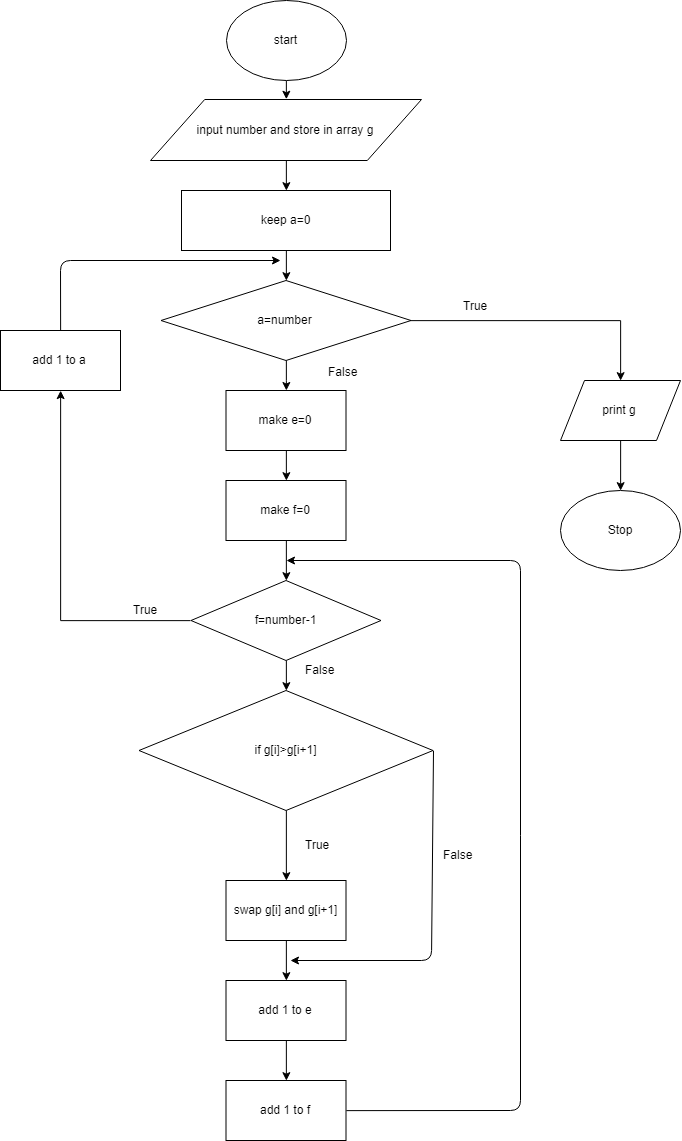
2.1Flowchart:**

Figure 1: flowchart

# 3.Data Structures

Data structures are a way to organize and store data in order to effectively access and work with it. It gives the relationship between the data, and the operations that can be carried out on the data. There are many different types of defined data structures that make it easier for both data scientists and computer engineers to focus on the main picture of solving larger issues rather than getting lost in the details of data description and access (Jose & Lal, 2016).There are two types of data structures they are primitive data structures and non-primitive data structures which are described below.

## 3.1 Primitive Data Structures

They are the data manipulation building blocks and contain pure, simple data values. Python has four different types of primitive variable:

## 3.1.1Integers

We can use an integer to represent numerical data, and more specifically entire numbers from negative infinity to positive infinity, such as 1,4, 5, or -1 (Pandovski, 2018).

## 3.1.2 Float

Float means the number of the floating point usually ending with a decimal figure like 11.11 ,10.44 or 13.14, it can be used for rational numbers (Pandovski, 2018).

## 3.1.3 String

Strings are alphabet, word or other character collections. In Python, you can create strings by enclosing in a pair of single or double quotes a sequence of characters. For instance: "gun”, “Girija”, ‘Tamang’ (Sturtz, 2018).

## 3.1.4 Boolean

This integrated data type that can take the values: True and False, which often allows the values 1 and 0 to be interchanged. Booleans are very useful in terms of conditionality and comparison (Sturtz, 2018).For example:

1. a=2, b=4 2.x=5, y=10

a==b y>x

false true

## 3.2 Non-Primitive Data Structures

The sophisticated members of the data structure family are non - primitive types. Not only do they store a value, but they collect values in different formats. The non - primitive data structures in the traditional computer science world are divided into:

## 3.2.1 List

Python lists are used to store collections of mixed items. These are mutable, which means that without changing their identity, we can change their content. Lists of elements separated by a comma can be identified with its square brackets [ and]. Python provides many methods for manipulating and working with lists. Normal list manipulations include adding new items to a list, removing items from a list, sorting or reversing a list. Python lists: you don't have to appeal them separately (Tagliaferri, 2016).

## 3.2.2 Tuple

Tuples are another standard type of sequence data which is similar to the list. The difference between tuples and list is that tuples are immutable, meaning you can't delete, add or edit any values within them once you have defined them. This may be useful in situations where you may pass control to someone else, but you don't want them to manipulate data in your collection, but rather see them or perform separate operations in a copy of the data. It can be considered as read only lists which are enclosed in parentheses () (Mester, 2017).

## 3.2.3. Set

Sets are a collection of unordered stuffs which are unique. These are useful for creating lists containing single data only. It's an unordered, but mutable collection which are very helpful for a large dataset to be processed. Since sets are unordered, there is no concept of index and specific order. They are denoted by {} (Jose & Lal, 2016).

## 3.2.4. Dictionary

Dictionary is an ordered collection of pairs of key value. Usually it is used when we have an enormous amount of data. We need to know the key to retrieve the value. In python, dictionaries are defined in braces {} with each item being a pair in the key: value. Key and value of could be if any type. Keys are usually numbers or string. While any arbitrary python object may be values. Dictionaries are enclosed with curly braces ({}) and square braces ([]) can be used to assign and access values (Jaiswal, 2017).

## 3.2.5. Array

First of all, Python arrays are a compact way to gather basic data types, all entries in an array must be of the same type of data. Unlike other programming languages like C++ or Java, however, arrays are not all that popular in Python. Arrays can be seen as a more efficient way to store a certain type of list for Python. However, this type of list has elements of the same type of data. The array module supports arrays in Python and you need to import them before you start initializing and using them. The data type of the elements stored in an array is restricted. During the creation of the array, the data type is specified using a code (Jaiswal, 2017).

There are some data about some movies:

|  |  |  |  |
| --- | --- | --- | --- |
| Movie ID | Movie Name | Rent Price | Quantity |
| M001 | Pulp Fiction | $5 | 30 |
| M002 | Lords of The Rings | $2.5 | 10 |
| M003 | The Revenant | $4 | 20 |

The above given data gives full information about movies name its rent price and quantity so I would like to use list data type for representing and storing the given data in python program because of the following reasons:

1.List is dynamic and mutable so the elements of it can be changed as user want.

2. Lists can contain arbitrary objects.

3.List can be sliced.

4.Lists are ordered.

5.List elements can be accessed by index.

6.Lists can be nested to arbitrary depth.

# 4.Learning Reflection

Truly speaking I had a basic knowledge of computer that I gained in my school level. In the beginning, programming was an unknown field for me. I wanted to be more familiar in this area so I had done many researches about different programming languages. As I was interested in this field of computing from beginning, therefore I had taken different classes of c programming, java, and php before coming to this module. I was performing research as I found that this module is all based on researching in my early bird classes of my college. I thought programming languages will be very easy for me as I have done all the classes before coming to this module.

This is the era of technology so, I was interested in this field from very beginning. I found there is a wide scope in the field of programming and computer related field. I wanted to be a good graphic designer and web page developer in near coming future. I was expecting to learn multiple programming languages develop websites. I was expecting that the teachers will not be familiar and it would be difficult for me to cope the requirements. I was not involved in this field before so, I thought it will be burden for me.

Without doubt I was able to meet the expectation in reality. When I entered the course, I found teachers cooperative and familiar. I found students polite and helpful which is a plus point for me. I was researching on websites and Youtube channels about python which made it easier for me to fight back module. I was able to develop simple website with the help of online pages. It was a very good learning experience in this collage. I think I will be able to achieve my aim of becoming a good programmer for graphic designer and web page developer.

This module contains variety of study materials to deal with which is not an easy task. Teachers were cooperative, so that they provided all the materials which is required to tackle the module. As this course is mainly based on researching, I performed high level of research to deal with the confusing which were created during study process. I thought python was easy programming language but I was wrong. It requires high amount of researching and cooperation with teachers to tackle module.

I find python programming language difficult to cope with. Bubble sort was also a hard concept for me to understand. The data structure is also a new concept for me to deal with. To deal with difficulties I am researching at next level. Now I am maintaining focus when I am studying. I think I will be successful in future after engaged in this field. Now I am able to create my own logic which make me crystal clear about the solution of any problems.

In the process of learning new things we assuredly face many difficulties. Similarly,

In the process of learning python programming languages, I had to suffer from many problems and difficulties. The main problem for learning python in this university is it is based on new learning process which is different from the learning area from where I came To overcome from this problem I am involving in researching .

# 5. References

Gupta, V., 2017. *Bubble Sort in Python with Example.* [Online]   
Available at: https://www.stechies.com/bubble-sort-python/  
[Accessed 31 march 2019].

Jaiswal, S., 2017. *Python Data Structures Tutorial.* [Online]   
Available at: https://www.datacamp.com/community/tutorials/data-structures-python  
[Accessed 02 April 2019].

Jose, J. & Lal, S., 2016. *Introduction to computing and problem solving with python.* 2nd ed. New Delhi: Khanna Book Publishing Co.(p)Ltd.

Mester, T., 2017. *Python Data Structures.* [Online]   
Available at: https://data36.com/python-data-structures-data-science-basics/  
[Accessed 02 april 2019].

Pandovski, Z., 2018. *Primitive Data Types in Python 3.* [Online]   
Available at: https://able.bio/ZoranPandovski/primitive-data-types-in-python-3--57tqcfp  
[Accessed 1 April 2019].

Parewa Labs Pvt. Ltd., 2018. *programiz.* [Online]   
Available at: https://www.programiz.com/dsa/bubble-sort  
[Accessed 31 march 2019].

Parewa Labs Pvt. Ltd., n.d. *programiz.* [Online].

Sehgal, K., 2018. *An Introduction to Bubble Sort.* [Online]   
Available at: https://medium.com/karuna-sehgal/an-introduction-to-bubble-sort-d85273acfcd8  
[Accessed 31 march 2019].

Sturtz, J., 2018. *Basic Data Types in Python.* [Online]   
Available at: https://realpython.com/python-data-types/  
[Accessed 1 April 2019].

Tagliaferri, L., 2016. *Understanding Lists in Python 3.* [Online]   
Available at: https://www.digitalocean.com/community/tutorials/understanding-lists-in-python-3  
[Accessed 02 april 2019].