Model Code & Module Titel

CC4002NA Information Systems

Assessment Weightage & Type

20% Individual Coursework

Year and Semester

2019 spring

Student Name: Bikram kumar Sharma

London Met ID:

Collage ID: np01cp4s190050

Assignment Due Date: 2019/04/05

Assignment Submission Date: 2019/04/05

I confirm that I understand my courseeork needs to be submitted online via Google Classroom 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.

Table of Contents

Table of figure

Table of tables

1. The Bubble Sort Algorithm

(unknown, n.d.)

Introduction:

Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent element if they are in wrong order. The swapping process goes on until and unless the number in the list are properly arranged sequence (unknown, n.d.)

How does Bubble sort works?

The bubble sort works on 3 steps first step, second step and third steps. The bubble sort algorithm also occupies less space than that of the other algorithm that we have learned.

Example to explain the bubble sort algorithm for ascending order:

Example:

First pass:

(**5 1** 4 2 8) **->** (**1 5** 4 2 8), Here, algorithm compare the first two element, and swaps since 5 > 1

(1 **5 4** 2 8) -> (1 **4 5** 2 8), Swap since 5 > 4

(1 4 **2 5** 8) -> (1 4 **2 5** 8), Swap since 5 > 2

(1 4 2 **5 8**) -> (1 4 2 **5 8**), Now, since these element are already in order (8 >5), algorithm does not swap them.

Second pass:

(**1 4** 2 5 8) -> (**1 4** 2 5 8), since these element are already in order (1 >4), algorithm does not swap them.

(1 **4 2** 5 8) -> (1 **2 4** 5 8), Swap since 4 > 2

(1 2 **4 5** 8) -> (1 2 **4 5** 8), since these element are already in order (5 >4), algorithm does not swap them.

(1 2 4 **5 8**) -> (1 2 4 **5 8**), since these element are already in order (8 >5), algorithm does not swap them.

Third pass:

Now, the array is already sorted, but our algorithm does not know if it is completed. The algorithm needs one whole pass without any swap to know it is sorted.

(**1 2** 4 5 8) -> (**1 2** 4 5 8), since these element are already in order (2 >1), algorithm does not swap them.

(1 **2 4** 5 8) -> (1 **2 4** 5 8), since these element are already in order (4 >2), algorithm does not swap them.

(1 2 **4 5** 8) -> (1 2 **4 5** 8), since these element are already in order (5 >4), algorithm does not swap them.

(1 2 4 **5 8**) -> (1 2 4 **5 8**), since these element are already in order (8 >5), algorithm does not swap them (unknow1, n.d.).

Output: (1 2 4 5 8), which is in ascending order.

Algorithms for bubble sort

Step 1: Take input a list (li)

Step 2: Declare a variable as sorted and value is false. The while loop is started until the value of the variable is false.

Step 3. New variable is created(x), and assign its value zero and let I equal 1.

Step 4. Start for loop with an index of position 'i' to the no. of element in the list.

Step 5. It is checks whether the value li [i-1] > li[i].

Step 6. Swap the value of li [I] and li [i-1] to create new variable.

Step 7. If the above condition is true then the value of i is increased by 1. This condition will run until the value of x is true and executed if false.

Step 8. Both if condition and for loops are ended.

Step 9. If loop starts again and checks if the x is equal to zero. If the value of x is equal to zero, then sorted will be true else sorted will be false.

Step 10. While loop is terminated.

Step 11. The entered loop is printed in ascending order.

Step 12. The bubble sort is ended.

2. Graphical representation for Bubble Sort algorithms:

Graphical representation of algorithms is known as Flowchart. As it contains various kinds of diagram and figure, which are joined by arrow head. Graphical representation for Bubble Sort algorithms is:

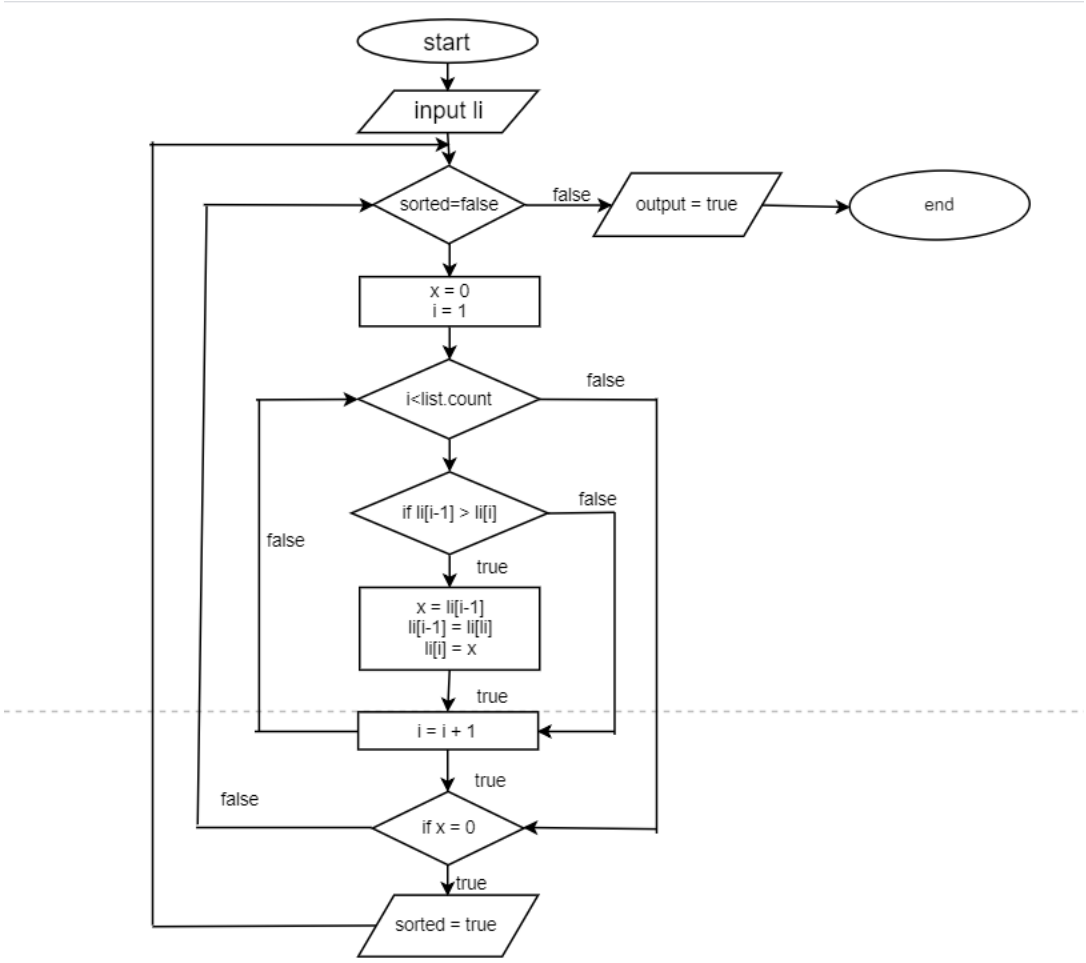

Fig. flowchart of Bubble Sort

3. Data structures

Data structures are basically just that-they are structures which can hold some data together. In other words, they are used to store a collect of related data. There are four built-in data structures in Python- list, tuple, dictionary and set.

A. List data Types:

List is an ordered sequence of items, accessible by index. It is one of the most used data type in python and is very flexible. A list is denoted by square bracket []. It contain elements usually of same type but also contain mixed types. List element can be changed so it is mutable. Index mainly starts from 0 to the remaining numbers present in the list. Each element in the list are separated by comma.

I. Len() function:

This function is used in list to calculate the number of items in a list.

```
A= [“h”, “e”, “l”, “l”, “o”]
Print (Len (A))
```

The output of given program is 5.

ii. Changing elements:

Hence the element in the list are mutable, their structure and content can be changed. Assigning to an element at an index changes the value.

```
L = [2, 1, 3]
L [1] = 4
Now, L= [2, 4, 3]
```

iii. Operation on list(add and remove):

In list we use two different operation to change element in the list i.e. add and remove. To add

elements in the list we use append and extend. We use append to add element to the end of list and extend for add list of element to the end. We use L .append () & L .extend () to add element respectively.

B. Tuples data Types:

Tuples are an ordered sequence of element and mix element types. Tuples once created cannot be changed and represented by parenthesis.

t = (1, 2, 3, “NJR”)

C. Dictionary data types:

A dictionary is an unordered collection of key-value pairs with unique keys. Dictionaries are mutable, so its items can be easily added/removed. Since dictionary are unordered, there is no concept of order/position, a definite order is not guaranteed. It is denoted by {}.

L = {‘id’: 1998, ‘name’: ‘Cap’, ‘size’: 3}

D. Set Data Types:

Set is an unordered collection of element. Set are mutable, so we can easily add or remove items, but since they are unordered, there is no concept of index, any specific order is not guaranteed. It always contain unique items, adding duplicate item is safe but pointless. It is denoted by {}.

(unknow2, n.d.)It is denoted by {}.

L = {1, 2, 3}

Data Table:

Movies ID	Movies Name	Rent Price	Quantity
M001	Pulp Fiction	$5	30
M002	Lord Of The Rings	$2.5	10
M003	The Revenant	$4	20

The value of above table can be stored more easily by using the list data types. As list is an ordered sequence of items, accessible by index. It is one of the most used data type in python and is very flexible. It contain elements usually of same type but also contain mixed types. List element can be changed so it is mutable. List also has a lot of function such as Len, append, pop, remove, etc. Reversing the values of the list can also be performed. Hence list data types has many features to store the data of above table in list in easy way than other data types.

Movies ID= [M001, M002, M003]

Movies name= [Pulp Fiction, Lord of the Rings, The Revenant]

Rent Price= [$5, $2.5, $4]

Quantity= [30, 10, 20]

We can also store the data from the above table by using 2 dimensional list (2D list).

4. Learning Reflection:

My preparation for the subject before I started the Module:

As I have no interest in computing or programming, I haven't have any programming ideas. Even I selects IT, I have dough on me that I got success or not. But When I came to my first lecture class of programming, I know the some newly words like python, coding, class, etc. But week to weeks pass I learn a lots of knowledge of python.

Expectation from the Module when I started it:

From the date of my admission, I used to study our course book and I have hurry to know about:

i. Describe the flowchart and algorithms,
ii. Describe the different loops.
iii. Describe different data types.
iv. Describe bubble sort.

Was I able to meet those expectation?

Yes, I was able to meet those expectations. As I have no knowledge of computer, from my first class I give more attention in lecture, tutorial and lab. Then I started to note in both in lecture and tutorial classes which help me more. As practical is more knowledgeable than theory, I was able to cod program. Day by day I was able to get knowledge in what I have expectation. Here, the Environment I have found is quite good for learning. Teacher staff are very communicated helpful.

How am I coping with the requirement and am I coping with it?

I'm coping quite well than I expect. All the required materials i.e. Tutorial, lecture and lab slides are uploaded timely, so it is not hard to do revision what had we taught during class. Also we research for our extra requirement.

What are my current difficulties and how can I overcome it?

At first, I think python is quite easy. But week to weeks I found difficulties in coding. I have lots of problems at first, because it was new for me, but by my hard labor I was able to solve it. My friends and lab, lecture and tutorial teachers helps me a lot in problems. Always I raise my hand when I didn't know in classes. My best way to solve any problem is I daily read slides of lecture, tutorial. Daily I practice my lab works and research in net.

Bibliography

unknow1, n.d. *freeCodeCamp Algorithm Bubble Sort Guide - The freeCodeCamp Forum.* [Online]
Available at: https://www.freecodecamp.org/forum/t/freecodecamp-algorithm-bubble-sort.../16097
[Accessed 05 04 2019].

unknow2, n.d. *Swift Fundamentals | Arrays and Sets - Cocoacasts.* [Online]
Available at: https://cocoacasts.com/swift-fundamentals-arrays-and-sets
[Accessed 05 04 2019].

unknown, n.d. *Bubble Sort | freeCodeCamp Guide.* [Online]
Available at: https://guide.freecodecamp.org/algorithms/sorting-algorithms/bubble-sort/
[Accessed 05 04 2019].

THANK YOU