Bahria University, Islamabad

Department of Software Engineering

Computer Programming

(Fall-2023)

Teacher: Dr. Raja Sulaiman

Student : Mustafa Dilawar Khan

Enrollment : 01-131232-074

Journal: 01

Task No:	Task Wise Marks		Documentation Marks		Total Marks (20)
	Assigned	Obtained	Assigned	Obtained	
1					
2					
3					
4					
5					

Comments:

Signature

Algorithms

Introduction

Algorithms are building blocks of computer programming. It is written in plain English, and any non-technical person can understand what is happening.

Tools Used

Microsoft Word and YouTube (for information about sorting).

Question 1: Finding the Shortest Path

Imagine you are developing a GPS navigation system. You are given a map with various locations and the roads connecting them. Your task is to write an algorithm to find the shortest path from one location to another. You can assume that you have a list of locations and the distance between each pair of locations. Your algorithm should output the shortest path and the total distance.

Algorithm:

Step1: Start

Step2: Store some pair of locations in the program with different routes and distances and this data set will act as our map.

Step3: Input the pair of locations he/she wants to visit by the user. [Input]

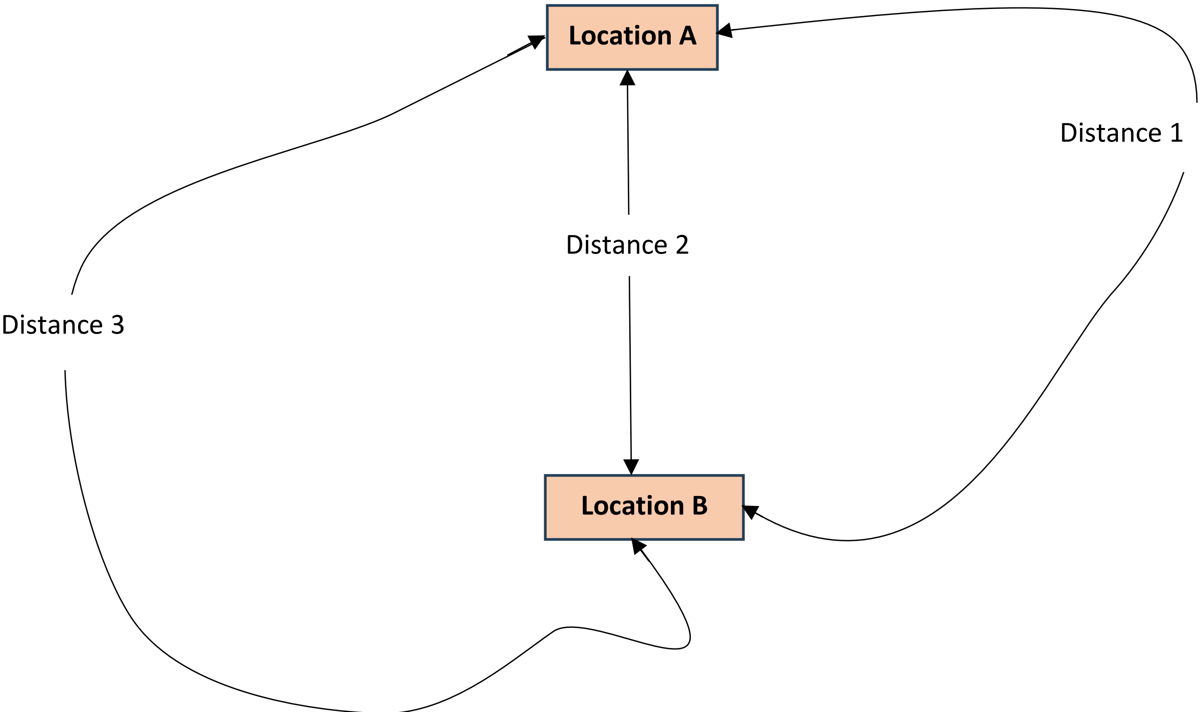
Step4: Then the program will search the user's entered pair of locations from the data set we have stored. [Process]

Step5: If the distance is found, then the set of same pair of locations with the different distances(map) is compared among each other. [Condition/Process]

Step6: Then the algorithm will show the shortest distance and print it for the user. [Output]

Step7: Stop.

Figure:



Question 2: Sorting a List of Numbers

You are working on a project where you need to sort a list of numbers in ascending order. Design an algorithm to efficiently sort a list of integers. You should consider various sorting algorithms, evaluate their time complexity, and choose the most suitable one for the task.

Algorithm:

Step1: Start

Step2: Input by the user of numbers and create a list of numbers. [Input]

Step3: After the list is created, the program will find the ascending order through **sorting methods**. [Function]

Step4: Print the list of numbers in ascending order. [Output]

Step5: Stop.

Explanation (Sorting Methods):

There are a lot of sorting methods, but I went through some of them and came to a conclusion about the most efficient method. So, here are some of them:

Bubble Sort:

A simple sorting algorithm that repeats steps through the list is bubble sorting. This process involves a lot of swapping, which uses a lot of memory and time.

Quick Sort:

Quick Sort is a divide-and-conquer sorting algorithm that is quite effective. After choosing a "pivot" element, the list is divided into two sub lists, each of which is then subdivided into additional sub lists.

Counting Sort:

A count Sort is an excellent sorting technique for short lists. It rapidly sorts the numbers by counting the number of times each one appears. It takes extra memory for the values that aren’t there in the list but reserves the memory for those values too.

Conclusion:

In my opinion, Quick Sort is the most efficient method for large list.

Question 3: Calculating Fibonacci Numbers

The Fibonacci sequence is a series of numbers where each number is the sum of the two preceding ones (e.g., 0, 1, 1, 2, 3, 5, 8, 13, ...). Write an algorithm to calculate the nth Fibonacci number. Your algorithm should be efficient and capable of handling large values of n.

Algorithm:

Step1: Start

Step2: Set 2 variables and assign them values like num1=0, num2=1 (first two Fibonacci Numbers).

Step3: Input number (required nth term of Fibonacci sequence) to find the nth Fibonacci number.
[Input]

Step4: As we are having the first two Fibonacci numbers, we will be using a loop to calculate the required Fibonacci number and the loop will be started from 3 to the number (nth term).
[Step4 -Step8=Processing]

Step5: Now we will calculate the Fibonacci number by adding num1 and num2, e.g., value=num1+num2

Step6: Now we will swap the values e.g., num1=num2

Step7: Then swap the value num2=value

Step8: The loop (Step5 – Step7) will be working until the desired nth Fibonacci number is calculated.

Step9: The final number which will be printed is num2 which will be the required nth Fibonacci number. [Output]

Step10: Stop

Question 4: Inventory Management

You are tasked with creating an algorithm for a store's inventory management system. Your algorithm should be able to add and remove items from the inventory, update the quantity of existing items, and generate reports of the items and their quantities. Design an algorithm that efficiently manages the store's inventory based on these requirements.

Algorithm:

Step1: Start

Step2: We will be creating an inventory list which will include the items already there and a place for new entries.

Step3: Input will be taken from the user and see if the item already exists then update the quantity otherwise add the item to the list. [Input]

Step4: If an item is having quantity ‘0’ or has reached a particular quantity limit, then you have two options, whether you want to remove it from the list or re-order it. [Processing]

Step5: The report of the updated items list will be generated and printed on the screen. [Output]

Step6: Stop.