



Bahria University, Islamabad

Department of Software Engineering

Computer Programming

(Fall-2023)

Teacher: Dr. Raja Sulaiman

Student : Syed Muhammad Saeed

Enrollment : 01-131232-086

Lab Journal:

Date:

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

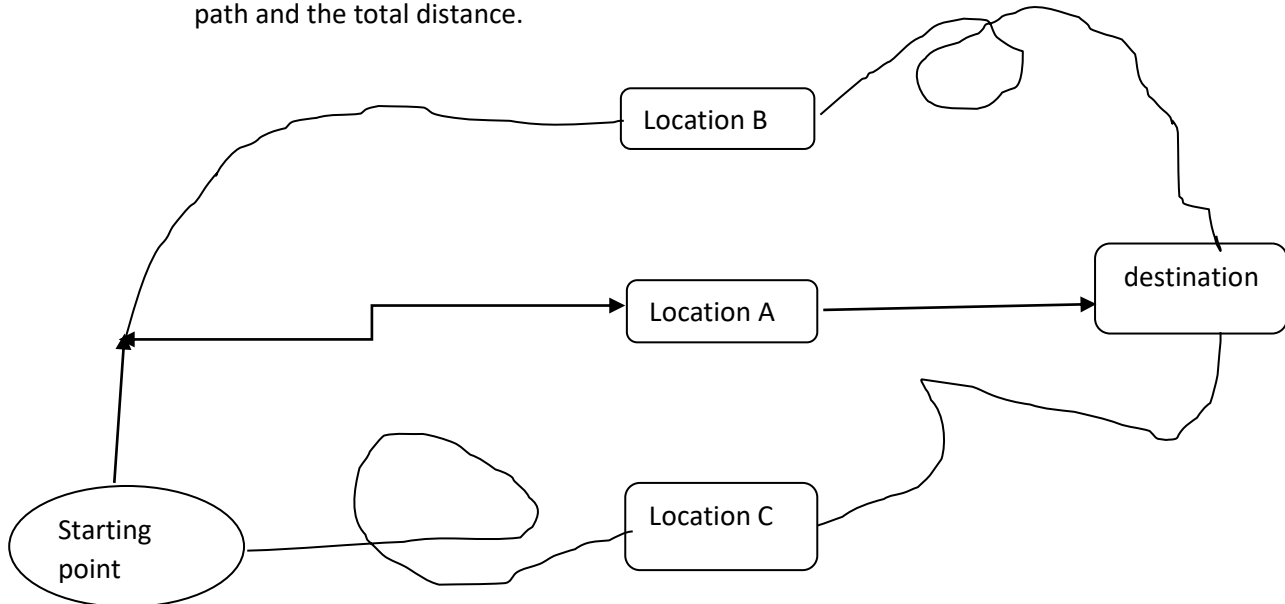
Comments:

Signature

- ALGORITHM

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:

START1: Start

STEP2: Traveler's Input

The trip starts when the traveler tells where they are and where they want to go.

STEP3: Map Setup

We possess a map containing various places and the roads that connect them, all set for exploration.(input)

STEP4: Search for the Pair

The algorithm begins by identifying the two locations selected by the traveler. (process)

STEP5: Exploring Options

Once the pair is located, it displays all the available routes along with their respective distance. (condition/process)

STEP6: Option for the Shortest Route

The aim of the algorithm is to find the shortest route ,securing and efficient journey.(output)

STEP7: Stop

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:

Suppose the list of number is(8,9,5,7,11,6,3,2,4,10,1)

Let Suppose our List is (8,9,5,7,11,6,3,2,4,10,1)

Step 1: Start

Step 2: Enter the List of Numbers. (8,9,5,7,11,6,3,2,4,10,1) {Input}

Step 3: Finding Ascending Order through Quick sorting method.
{Function}

Step 4: Print the Array. {Output}

Step 5: End.

Explanation:

Different Sorting method.

QUICK SORT:

Quick sort is the ideal for this list. It boasts an average case time complexity of $O(n \log n)$ and tends to work well with various types of data. Quick sort is especially efficient when dealing with larger lists, and its worst-case scenario is less likely to happen with random input lists.

MERGE SORT

Merge sort is another excellent choice . It maintains a consistent time complexity of $O(n \log n)$, making it efficient for sorting lists, regardless of how they are initially ordered. Merge sort is also stable ,which means it preserves the relative order of equal elements ,a useful feature in certain situations.

BUBBLE SORT

Bubble sort is not a good choice for this list; especially if it's a large list. It can have a worst-case time complexity of $O(n^2)$ and is generally less efficient than quick sort or merge sort.

So, the best sorting from the given list is Quick sort. As it will be more efficient and flexible for various input scenarios.

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:

Get the number 'n' which tells us which Fibonacci number you want to find.

STEP3:

Create two variables, we will name them 'a' and 'b', and set their values to 0 and 1, which represent the first two Fibonacci numbers.

STEP4:

Begin a loop starting from the third position up to the nth position for each position ranging from 3 to 'n', follow steps 5 through 7.

STEP5:

Find the next Fibonacci number, denoted as 'c', by adding 'a' and 'b' together ($c=a+b$).

STEP6:

Update 'a' to hold the previous values of 'b'.

STEP7:

Update 'b' to be the former values of 'c'.

Step8:

When the loop concludes, 'b' will contain the Fibonacci number corresponding to the nth position.

Step 9:

Print the value of 'b'.

Step 10: 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 are making a list of things we have and we are also making space for adding new things.

STEP3:

We'll ask the user for input, and if the item is already on the list, we will change the quantity. if it is not there, we will put it on the list .

STEP4:

If an item has a quantity of '0' , you can choose between two option ,either remove it from the list or re-order it.

STEP5:

We'll create a report with the updated list of items and it will be displayed and printed on the screen.

STEP6: Stop