

Group members:

Member 1: Waleed Ahmed Razzaqi
Enrollment number: 01-131232-093

Member 2: Malik Shajar Hussain
Enrollment number: 01-131232-043



Computer Programming CSC-113

Assignment 1: Problem Solving (CLO-2)

Group members:

Member 1: Waleed Ahmed Razzaqi
Enrollment number: 01-131232-093

Member 2: Malik Shajar Hussain
Enrollment number: 01-131232-043

Group members:

Member 1: Waleed Ahmed Razzaqi
Enrollment number: 01-131232-093

Member 2: Malik Shajar Hussain
Enrollment number: 01-131232-043

Question 1: Finding the Shortest Path:

1. Start
2. Let us consider a marked graph having Single sources u and path vertices as A, B, C, and D, D be our destined vertex.
3. The initial vertex is declared as a single source S having a distance $d(s)=0$.
4. The distance from S to A, and S to C are directed paths and S to B, and S to D are via paths.
5. The path from the source to all other vertex has not been determined yet, so we use the infinity symbol to represent this initially.
6. For 1st path Join loc S to A. Now we know the location of s is at distance $d(s)=0$ and weight let's suppose from vertex S to A will be x so now when will move from S to A mathematically we will implement $D(s) + C(s, A) = D(A)$ which is $0+x<\text{infinity}$ so we will initialize the distance from S to A as x and vice versa it will happen to any path between two vertices.
7. So our first path will be from S to A, A to B, and B to D and the cost of each vertex will be added as our single source moves from point S to D through this path.
- 8, So our second path will be from S to C, and C to D, and the cost of each vertex will be added as our single source moves from point S to D through this path.
9. So our third path will be from S to C, C to B, and B to D and the cost of each vertex will be added as our single source moves from point S to D through this path.
10. Now we will compare all the costs stored in each path taken by the source S.
11. The smallest value will act as a result of the shortest path S to D.
12. Stop

Group members:

Member 1: Waleed Ahmed Razzaqi
Enrollment number: 01-131232-093

Member 2: Malik Shajar Hussain
Enrollment number: 01-131232-043

Question 2: Sorting a List of Numbers:

1. Start
2. Let us declare an array of certain values.
3. Let us consider a pivotal value in those certain values.
4. We will initialize the lowest value of the array as i and the highest value of the array as j .
5. We will initialize the highest at infinity because we are not aware of it and will be the highest value of the array.
6. i will be a value greater than the pivotal point and j will be a value lower than the pivotal value.
7. Increment the i value by 1 and decrement the j value by 1 and check the simultaneous value which satisfies step 6 of the algorithm.
8. if both the values of i and j satisfy step 6 then swap the values.
9. Perform this procedure until j holds the position of the pivotal point and the values of the array will be sorted in ascending order after the pivotal point.
10. The average time complexity of quick sort is $O(N \log(N))$.
11. Stop

Group members:

Member 1: Waleed Ahmed Razzaqi
Enrollment number: 01-131232-093

Member 2: Malik Shajar Hussain
Enrollment number: 01-131232-043

Question 3: Calculating Fibonacci Numbers:

Step 1. Start

Step 2. Let us declare 5 variables the as f1,f2,f3,n,I,

Step 3. We will set the value of the following variables as f1=0,f2=1,f3=0, I=1,n.

Step 4. Enter the range of numbers that you want to include in the Fibonacci series.

Step 5. Enter the value of n.

Step 6. Show the values of f1,f2.

Step 7. While I is greater than n-2, so loop will be executed if the statement is true.

then add the value of f1+f2 into f3 and extract f3 from it then f1 will be equal to f2 and f2 will be equal to f3. Increment I by 1 until the looping executes according to the condition

Step 8. Stop

Group members:

Member 1: Waleed Ahmed Razzaqi
Enrollment number: 01-131232-093

Member 2: Malik Shajar Hussain
Enrollment number: 01-131232-043

Question 4: Inventory Management:**Algorithm for Inventory Management:**

1. Start
2. Create a database or data structure to store inventory items.
3. Provide a user interface with the following options:
 - Add an item
 - Update quantity
 - Delete an item
 - Generate a report
 - Exit
4. If the user selects "Add an item":
 - Prompt the user to input the item name and quantity.
 - Store the item in the database with a unique identifier (e.g., product ID).
 - Return to the main menu.
5. If the user selects "Update quantity":
 - Prompt the user to input the item name.
 - Check if the item exists in the inventory.
 - If it exists, allow the user to update the quantity for that item.
 - If it doesn't exist, inform the user and return to the main menu.
6. If the user selects "Delete an item":
 - Prompt the user to input the item name.
 - Check if the item exists in the inventory.
 - If it exists, allow the user to delete the item and remove it from the database.
 - If it doesn't exist, inform the user and return to the main menu.
7. If the user selects "Generate a report":
 - Display a report listing all items and their current quantities in the inventory.
 - Return to the main menu.
8. If the user selects "Exit," end the program.
9. Repeat from Step 3 until the user chooses to exit.
10. Stop