National University of Computer and Emerging Sciences, Lahore Campus

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C:\Users\saif\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Word\final design.jpg | **Course:** | **AI Lab** | **Course Code:** | **CL 1002** |
| **Program:** | **BS (Data Science)** | **Semester:** | **Spring 2023** |
| **Duration:** | **90 Minutes** | **Total Marks:** |  |
| **Paper Date:** | **28-March-23** | **Weight** | **25 %** |
| **Section:** | **BDS-6A** | **Page(s):** |  |
| **Exam:** | **Mid** | **Reg. No.** |  |

**Instruction/Notes:** Honesty always gives fruit and Dishonesty is always harmful.

Notes:

1. Please attempt your Exam on Google Collab, no other tabs should be open, in such case your submission will not be considered, and you will be awarded with zero.
2. No Cheat sheet is allowed.
3. The Internet is not allowed.
4. Write appropriate main function and call your functions so, I can run and see your output.
5. There will be binary checking.
6. Submit a single .ipnab file with your roll No.

**Question No 1:**

Given a directed graph represented as an adjacency list, write a Python function **bfs\_with\_distance(adj\_list, start, num\_vertices**) that performs a BFS traversal starting from a specified vertex and returns the visited vertices in the order in which they were visited. Additionally, the function should return the distance of each visited vertex from the starting vertex.

The function should take as input the following parameters:

* **‘adj\_list’:** a list of lists representing the adjacency list of the graph. Each element of the list corresponds to a vertex in the graph and contains a list of the neighboring vertices.
* **‘start’:** an integer representing the starting vertex.
* **‘num\_vertices’:** an integer representing the number of vertices in the graph.

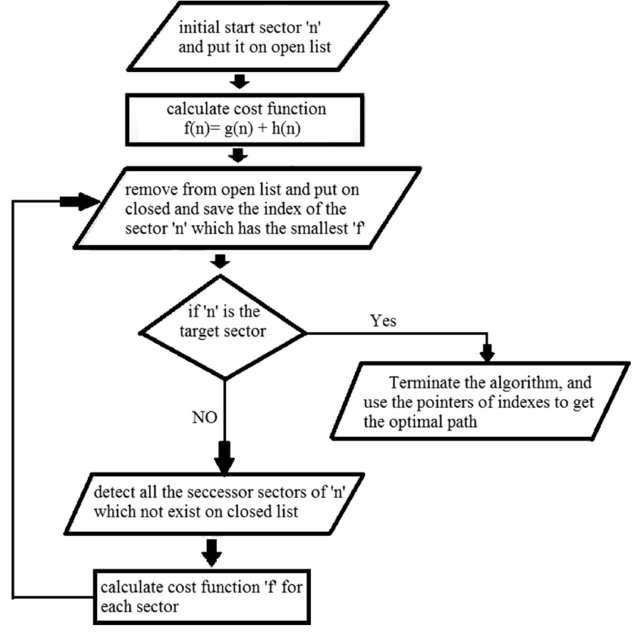
The function should return a tuple (**‘order\_visited’, ‘distances’**), where:

* **‘order\_visited’:** a list of integers representing the visited vertices in the order in which they were visited during the BFS traversal.
* **‘distances’:** a list of integers representing the distance of each visited vertex from the starting vertex. The distance of the starting vertex from itself is 0, and the distance of any unreachable vertex is **‘inf.’**

In addition to returning these values**, the function should also display the BFS path between two nodes. You can use the any functions from the Plotly library or Matplotlib to create the graph visualization**. You can assume that the vertices are labeled from 0 to **‘num\_vertices-1’**, where **‘num\_vertices’** is the number of vertices in the graph. You can also assume that the graph is connected, i.e., there is a path between any two vertices in the graph.

**Q2. Write a program to implement the A\* algorithm for finding the shortest path from a source node to a destination node in a weighted graph.**

**Explanation**: The A\* algorithm is used to find the shortest path from a source node to a destination node in a weighted graph. The algorithm works by using a heuristic function to guide the search and reduce the number of nodes explored. The time complexity of the algorithm is generally lower than Dijkstra's algorithm, but it depends on the specific heuristic function used.



Text

Description automatically generated