Project Altair

**Software and Automation Team Recruitment Batch 21**

****

**Prepared By:**

Khalid Hasan Ador

ID: 210042102

**LOGICAL PART**

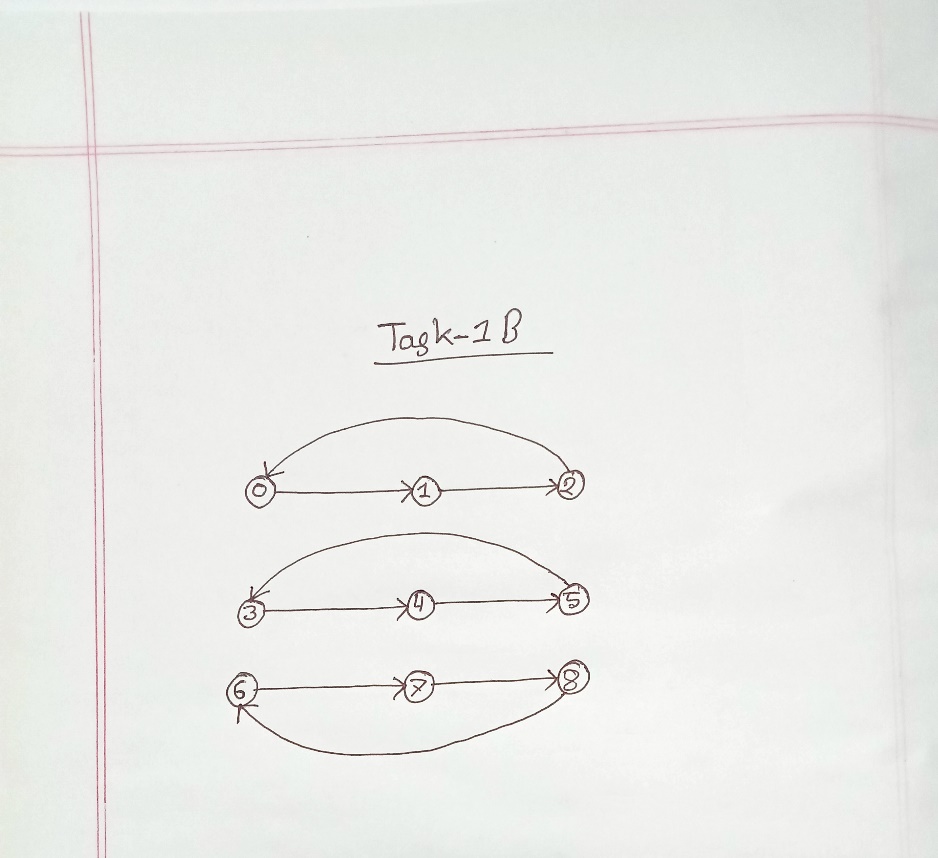
**Task-1A**: link dite hobe

**Explanation:**

The task was to find if there is a valid path to visit all the location. There exists a valid path if and only if the graph is connected. That is what I checked in my code. If the graph is connected than there will be only one connected component. So after taking all the inputs I checked the number of connected components. If there is only 1 such component than the code will output “TRUE” else it will output “FALSE”.

I used DFS (Depth First Search) to do the job. The number of connected component is simply the number of times dfs function will be called. So increase the ct variable each time dfs is called.

**Task-1B:**



**Explanation:**

From the given data there are 9 vertices and 9 edges. In a bi-directional graph an edge between vertex A and vertex B means there is an edge from A to B.

**Task-2:**

1. Dijkstra's Algorithm:

Sample Code Snippet: link dite hobe

Use Cases:

1. To find the shortest path
2. In social networking applications
3. In a telephone network
4. To find the locations in the map

**Advantages of Dijkstra’s Algorithm**

* It is used in Google Maps.
* It is employed to identify the shortest path.
* It is very popular in the Geographical Maps.
* It is used to locate the points on the map that correspond to the graph’s vertices.
* In order to identify the Open Shortest Path First, it is needed in IP routing.
* The telephone network makes use of it.

**Disadvantages of Dijkstra Algorithm**

* It conducts a blind scan, which takes a lot of processing time.
* It is unable to manage sharp edges. As a result, acyclic graphs are produced, and the ideal shortest path is frequently impossible to find.
* Calculates the distance from only one source vertex to other vertices

1. Bellman Ford Algorithm

Sample Code Snippet: link dite hobe

Use Cases:

1. For calculating shortest paths in routing algorithms
2. For finding the shortest path
3. During chemical reaction

Advantages of Bellman Ford Algorithm:

1. It will work for negative edges and negative cycles
2. It can detect negative cycles
3. It guarantees to find the shortest path in a graph with no negative weight cycles.

**Disadvantages of** Bellman Ford **Algorithm**

* It has a worst-case time complexity of O(V\*E), which is slower than Dijkstra's algorithm.
* It may not terminate if the graph contains a negative weight cycle.

**3)**  **Floyd Warshall Algorithm**

Sample Code Snippet: link dite hobe

**Use Cases:**

Advantages of **Floyd Warshall** Algorithm:

1. Can find the shortest path between all pairs of nodes
2. Time complexity of O(v^3) is ideal
3. Can detect negative cycles

Disadvantages of **Floyd Warshall** Algorithm:

1. Works only on graphs with couple hundred Vetices
2. Cannot handle negative weight cycle

4) A\* ( A Star ) Algorithm:

Use Cases:

1. Find the shortest path if the graph is dense
2. Robotics, video games, route planning, logistics, and artificial intelligence

Advantages of A\* Algorithm

* It guarantees finding the optimal path when used with appropriate heuristics.
* It is efficient and can handle large search spaces by effectively pruning unpromising paths.
* It can be easily tailored to accommodate different problem domains and heuristics.
* A\* is flexible and adaptable to varying terrain costs or constraints. Additionally, it is widely implemented and has a vast amount of resources and support available.

Disadvantages of A\* Algorithm:

* A\* can be computationally expensive in certain scenarios, especially when the search space is extensive and the number of possible paths is large.
* The algorithm may consume significant memory and processing resources.
* A\* heavily relies on the quality of the heuristic function. If the heuristic is poorly designed or does not accurately estimate the distance to the goal, the algorithm's performance and optimality may be compromised.
* A\* may struggle with certain types of graphs or search spaces that exhibit irregular or unpredictable structures.

Conclusion:

BFS is the basic algorithm to find the shortest path. If we have a weighted graph than we can use Dijkstra Algorithm. If we have negative weight in a graph, we should use Bellman Ford Algorithm. If we nedd all pair shortest path, we can use Floyd Warshell Algorithm. Lastly if the graph is too dense we can use A star search Algorithm instead of Dijkstra Algorithm.

**Theoretical Part**

Task-1:

I will use a combination of microcontrollers and a single-board computer.

A single-board computer (SBC) is a complete computer built on a single circuit board, with microprocessor(s), memory, input/output (I/O) and other features required of a functional computer. It can easily maintain the complex tasks of the robot like path planning.

On the other hand microcontrollers will be used to do specific low-level functions and real-time control of the hardware components.

REFERENCE:

1. <https://www.youtube.com/watch?v=F3PNsWE6_hM>
2. <https://byjus.com/gate/dijkstra-algorithm-notes/>
3. <https://www.youtube.com/watch?v=LKfIjVZ6kg4>
4. <https://www.youtube.com/watch?v=5MEZXOyaRmk>
5. <https://codereview.stackexchange.com/questions/230657/implementation-of-a-algorithm-in-c>
6. <https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/a-star-algorithm#:~:text=It%20is%20a%20searching%20algorithm,can%20find%20its%20own%20course>.