Algos & DS

Project 3

Data Structures

1. Linked List

For detecting cycle using back edging technique.

1. Arrays

For all the other operations.

Operations Performed on this code:

1. Create a Directed Graph.
2. Perform DFS Traversal.
3. Perform BFS Traversal.
4. Find Shortest Path.
5. Detect a Cycle.
6. Check if Graph a Bipartite.
7. Check if Graph a tree.
8. Exit.

ASSUMPTIONS: i. The start node is considered as root node.

ii. If no of vertices= 6, then the vertices are to be named between 0 and 5.

1. Create a Directed Graph.

The No of vertices, edges; Source, Destination and Weights are received as inputs and stored as matrix.

Eg: If a graph has a edge between 1 and 2 with weight=3, then it will be stored as G[1][2]=3.

1. DFS Traversal

Depth First Search is done by traversing an unvisited adjacent node and proceeding DFS traversal from that new node.

If a node does not have any adjacent node then backtrack to the previous node and continue DFS search.

1. BFS Traversal:

Breadth First Search is done by traversing through all the adjacent vertices that are unvisited and then proceeding with the next element in the array/queue.

1. Shortest Path (Dijkstras).

Steps:

1. Start from root node.
2. RELAX its adjacent nodes. ie. If the minimum cost to reach the current node from source node is less than its present value, then update the nodes cost value.
3. Go to next adjacent node with least cost.
4. Detecting Cycle

Steps:

1. Start from the source node.
2. Find back edges for that node by backtracking its connected nodes until start node is reached.
3. If current node and the backtracked node are the same then cycle is formed else proceed with next node.
4. Graph is Bipartite:

If all the adjacent nodes of a graph are of different colours then the graph is bipartite.

There can be only two colours red and blue.

In my code red means flag=1 and blue means flag=2;

1. Check if Graph is a tree.

For this there are three conditions to be checked.

1. Check for cycle
2. Check whether the node is a root node ie. No incoming edges.
3. Check whether the root node can traverse through all the vertices in the tree. This was implemented by traversing the tree from the root node.