

Algorithm Lab (Course Code: MC504)

Assignment - 6

Submission Deadline: within class timing, (17/02/2023)

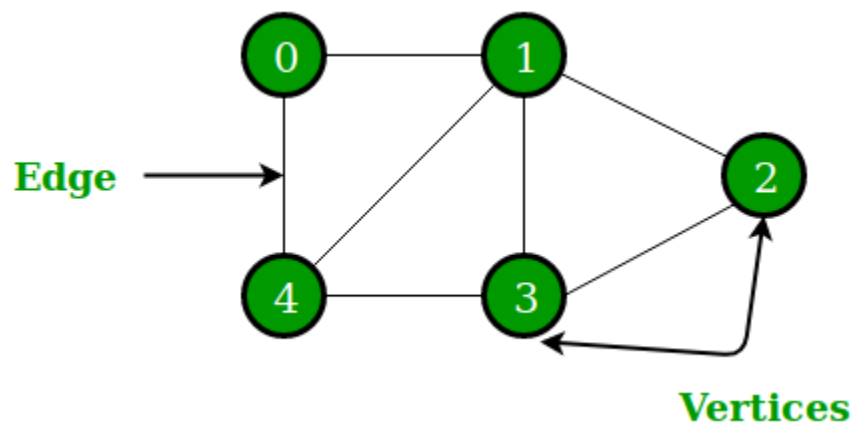
Total Marks: 20

Instructions:

- Proper indentation is mandatory.
 - Program files **must** be compiled using **linux gcc compiler**.
 - **VERY IMPORTANT:** You must add comments whenever necessary, to make the code understandable.
 - Markings will be based on the correctness and soundness of the outputs. Marks will be deducted in case of plagiarism.
 - Take inputs from users. Make necessary assumptions if required.
 - **ANSWER FILE:** Source code: (file name) e.g. A6_Q1.c
-

Q1.

A Graph is a non-linear data structure consisting of nodes and edges. The nodes are sometimes also referred to as vertices and the edges are lines or arcs that connect any two nodes in the graph.



In the above Graph, the set of vertices $V = \{0,1,2,3,4\}$ and the set of edges $E = \{01, 12, 23, 34, 04, 13\}$.

Now with this basic idea implement bi-directional graph data structure. Use the following format for Nodes:

```
struct node {
```

```
    int vertex;  
    struct node* next;  
};
```

And for Graph use the following:

```
struct Graph {  
    int numVertices;  
    struct node** adjLists;  
    int* visited;  
};
```

Helper functions to implement the graph data structure:

`struct node* createNode(int)` : Creates a node/vertex of graph having the value passed as integer parameter.

`struct Graph* createGraph(int n)` : Creates a graph with n nodes which is passed as parameter.

`void addEdge(struct Graph* graph, int src, int dest)` : For adding edges of graph. Here you will require the help of `createNode(int)`.

`void printGraph(struct Graph* graph)` : For visualization of graph. You can use adjacency list representation for visualization of graphs.

Now once the graph is constructed, you have to determine if there is a valid path that exists from vertex `Source_node` to vertex `Destination_node`. You have to take from user two inputs i.e. `Source_node` and `Destination_node` and then return true if paths exist otherwise false.

HINT: Design, Develop and Implement these functionality as menu driven Program for end user.

All the input should be user defined. For determining valid paths use either Breadth First Search(BFS) or Depth First Search(DFS) algorithm using either Queue or Stack data structure.

Make necessary assumptions as comments wherever required.