Viva Question with solution

Graph Data Structure

1. **Can you tell how linear data structures differ from non-linear data structures?**
   Answer:

- If the elements of a data structure result in a sequence or a linear list then it is called a linear data structure. Whereas, traversal of nodes happens in a non-linear fashion in non-linear data structures.
- Lists, stacks, and queues are examples of linear data structures whereas graphs and trees are the examples of non-linear data structures.

2. **What are the applications of graph data structure?**

   Graphs are used in wide varieties of applications. Some of them are as follows:

1. Social network graphs to determine the flow of information in social networking websites like facebook, linkedin etc.
2. Neural networks graphs where nodes represent neurons and edge represent the synapses between them
3. Transport grids where stations are the nodes and routes are the edges of the graph.
4. Power or water utility graphs where vertices are connection points and edge the wires or pipes connecting them.
5. Shortest distance between two end points algorithms.

3. **How do you represent a graph?**

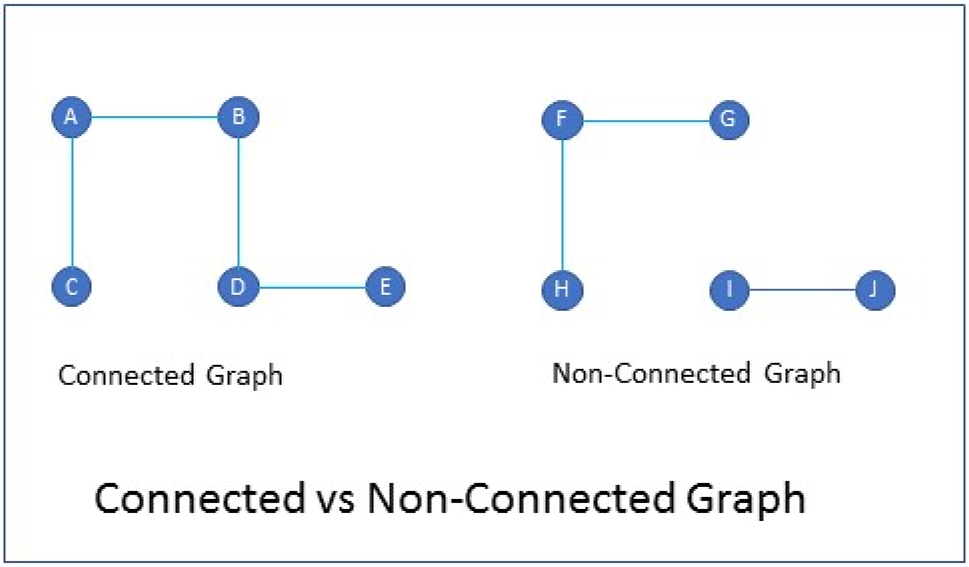
   We can represent a graph in 2 ways:

- Adjacency matrix: Used for sequential data representation
- Adjacency list: Used to represent linked data

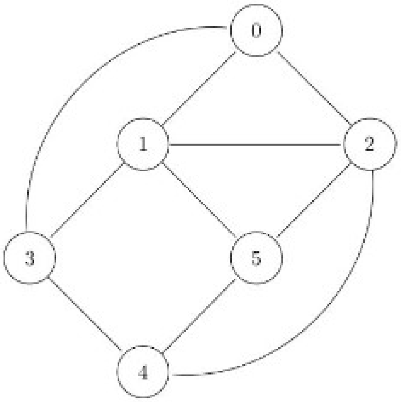
4. **What is the difference between connected graph and non-connected graph?**

   In a connected graph there is at-least one path from every vertex to every other vertex.

   In a non-connected graph every vertex may not be connected to every other vertex



5. What will be the sequence corresponds to that of **depth first search** for the graph given below.

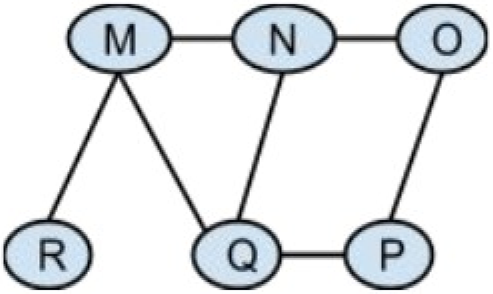


Answer:

A correct sequence of DFS traversal is 0 1 2 4 3 5

There is no edge between 2 and 3 and also 5 which is connected to 2 is unvisited.

6. Consider the following graph.



If we run **breadth first search** on this graph starting at any vertex, what will be the possible order for visiting the nodes ?

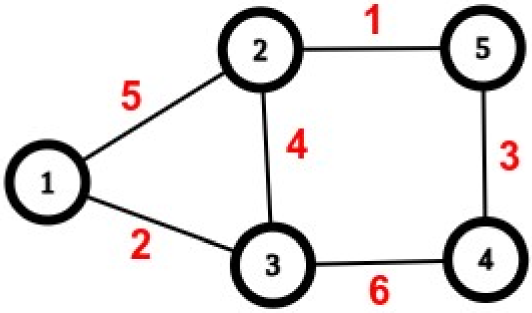
Answer:
QMNPRO

7. **How Many Spanning Trees Can A Graph Has?**

   It depends on how connected the graph is. A complete undirected graph can have maximum nn-1 number of spanning trees, where n is number of nodes.

8. **How to Find Minimum Spanning Tree Using Prim's Algorithm:**

   Find the weight of the MST



Answer: 10

9. Difference between Prim's and kruskal's algorithm with respect to time complexity?

10. What is the difference between Dijkstra's and BFS? And then why are the time complexities of these algorithms so different?

- **Breadth-first search** is just Dijkstra's algorithm with all edge weights equal to 1.
  - BFS basically just expands the search by one "step" (link, edge, whatever you want to call it in your application) on every iteration, which happens to have the effect of finding the smallest number of steps it takes to get to any given node from your source ("root")
  - Breadth-first search can be viewed as a special-case of Dijkstra's algorithm on unweighted graphs, where the priority queue degenerates into a FIFO queue.
  - Operations on a regular queue are `O(1)`.
  - BFS runs in `O(E+V)`.

- **Dijkstra's** algorithm is conceptually breadth-first search that *respects* edge costs.
  - For example, in routing the distances (or weights) could be assigned by speed, cost, preference, etc.
  - Dijkstra's uses a priority queue data structure to keep track of the frontier of unvisited nodes.
  - Operations on a priority queue are `O(log n)`.
  - Dijkstra's runs in `O((V+E)*log(V))`