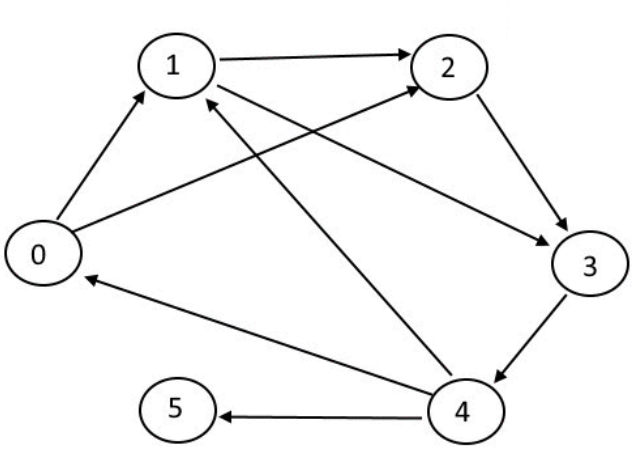
Military Institute of Science and Technology
Computer Science and Engineering
CSE-204, Week 11 - Implementation of Graph and BFS



https://algorithms.tutorialhorizon.com/breadth-first-searchtraversal-in-a-graph/

1. Implement the following graph data-structure.

```
class Graph
{
    int ** graph;
public:
    int vertices;
    int edges;
    Graph(int v);
    ~Graph();
    void addEdge(int u, int v);
    int * getAdjacentVertices(int u);
};
```

2. Implement a Circular Queue with the following skeleton.

```
class Queue
{
    int * q;
    //declare additional variables
public:
    Queue(int queue_length);
    void enqueue(int val);
    int dequeue();
    int isEmpty();
};
```

3. Implement the following BFS pseudocode.

```
BFS (G, s)                  //G = graph and s = source node
    let Q be queue.
    Q.enqueue(s)            //Inserting s in queue until all
                            //its neighbour vertices are marked.
    mark s as visited.
    while (Q is not empty)
      //Removing that vertex from queue,
      //whose neighbour will be visited now
      v  =  Q.dequeue()
      //processing all the neighbours of v
      for all neighbours w of v in Graph G
          if w is not visited
              Q.enqueue(w)       //Stores w in Q to further
                                 //visit its neighbour
              mark w as visited.
```

Source: https://www.hackerearth.com/practice/algorithms/graphs/breadth-first-search/tutorial/

4. Print the BFS traversal of the graph shown in the first page.
5. Modify the algorithm in (3) to implement the following pseudocode.

```
BFS(G, s)
 1  for each vertex u ∈ V[G] − {s}
 2       do color[u] ← WHITE
 3          d[u] ← ∞
 4          π[u] ← NIL
 5  color[s] ← GRAY
 6  d[s] ← 0
 7  π[s] ← NIL
 8  Q ← ∅
 9  ENQUEUE(Q, s)
10  while Q ≠ ∅
11       do u ← DEQUEUE(Q)
12          for each v ∈ Adj[u]
13               do if color[v] = WHITE
14                     then color[v] ← GRAY
15                          d[v] ← d[u] + 1
16                          π[v] ← u
17                          ENQUEUE(Q, v)
18          color[u] ← BLACK
```

6. Replace the adjacency matrix in (1) with adjacency list. (Use linked list)