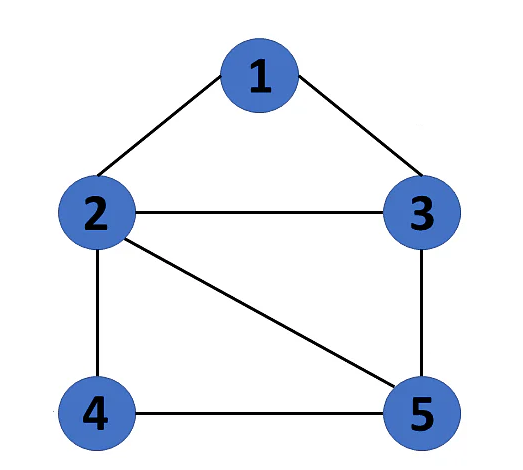
**Lab 4 Instructions**

Consider the following graph.



In today’s assignment you will perform a BFS on the graph. Your program should accept the graph as an adjacency matrix. Your program should also be able to take any node as the initial node and any other node as the goal node. At each stage you will have to output the node lists present in the BFS queue and the total number of nodes in the queue. The total number of nodes will be the sum of number of nodes in each list in the queue. The successor() function should have a node as a parameter and look into the adjacency matrix to find all nodes that are connected to it. Assume that the cost of all edges is equal. Your code need to print the optimal path from initial to goal nodes. Note that you will be graded *only for printing the contents of the queue and the final least cost path.*

The pseudocode of BFS is given below for your ready reference:

Create a queue that will store path(s) (of type list preferably)

Initialize the queue with the first path starting from *initial* state

Now run a loop till queue is not empty

get the frontmost path from queue

check if the lastnode of this path is goal node

if true then print the path

run a loop for all the vertices connected to the current node i.e. lastnode extracted from path

if the vertex is not visited in current path

a) create a new path from earlier path and append this vertex

b) insert this new path to queue

**Please show your work, even if it is partial, since every lab is being graded and grades will be awarded based on what you show during the lab.**