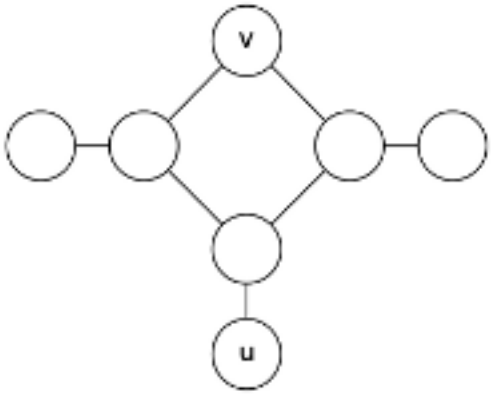
IT251 – DSA II

Assignment – 3

Name: Niraj Nandish

Roll No: 191IT234

Initially my algorithm consisted of doing BFS from a random node, store the last node then do BFS from that node and calculate the diameter. While this algorithm worked but only for acyclic graphs. Hence it failed for a few cases where graphs contain a cycle. For example:



To solve this issue while maintaining the time complexity, I made a list which stores all nodes having the diameter as the maximum diameter after the first BFS. Now, the BFS was done on every node in that list instead of just one node like last case. This method increased the time complexity but worked for most cyclic cases. Yet this algorithm failed in one case.

To maintain the deterministic of the algorithm, the final algorithm was devised which consists of doing BFS on every node and finding the longest-shortest path, the longest of all the shortest paths between any two nodes, of every node and find the maximum of all the longest-shortest path, which then becomes the diameter of the graph. The time complexity of this algorithm is $O(V(V+E))$. To maintain the space complexity list was used as a queue structure which saves space. Also, vectors were used instead of arrays wherever required to save space.