

LP-II Assignment No-1

• TITLE : DFS and BFS

• PROBLEM STATEMENT :

Implement depth first search algorithm and breadth first search algorithm. Use an undirected graph and develop a recursive algorithm for searching all the vertices of a graph or tree data structure.

• LEARNING OBJECTIVES :

1. To understand searching algorithms - depth first and breadth first
2. To make use of recursive function while implementation.

• S/W AND H/W REQUIREMENTS :

1. S/W - Pycharm (Python IDE), Python
2. H/W - 64-bit Windows 10 OS

• THEORY :

1. Undirected Graph

- An undirected graph is a set of nodes and a set of links between the nodes. Each node is called a vertex, each link called an edge, and each edge connects two vertices. The order of the two connected vertices is unimportant. An undirected graph is a finite set of vertices together with a finite set of edges.

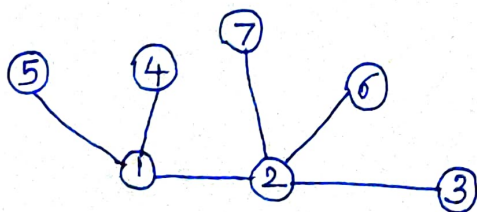
2. BFS

- Breadth first search for a graph is similar to breadth first traversal of a tree. Unlike trees, graphs may contain cycles, so we may come to the same node again. To avoid processing a node more than once, we use a boolean visited array. For simplicity, it is assumed that all vertices are reachable from the starting vertex.

3. DFS

- Depth first search for a graph is similar to depth first traversal of a tree. To avoid processing a node from than once, we used a boolean visited array. The algorithm starts at the root node and explores as far as possible along each branch before backtracking. The idea is to start from the root and mark the node and move to adjacent unmarked node and continue this loop until there is no unmarked adjacent node. Then backtrack and check for other unmarked nodes and traverse them. Finally print the nodes in path.

eg :- Consider the following graph -



BFS : 1, 2, 4, 5, 7, 3, 6

DFS : 1, 2, 3, 6, 7, 4, 5

- CONCLUSION:

Hence we implemented depth first and breadth first searching algorithms for an undirected graph using recursive functions.
