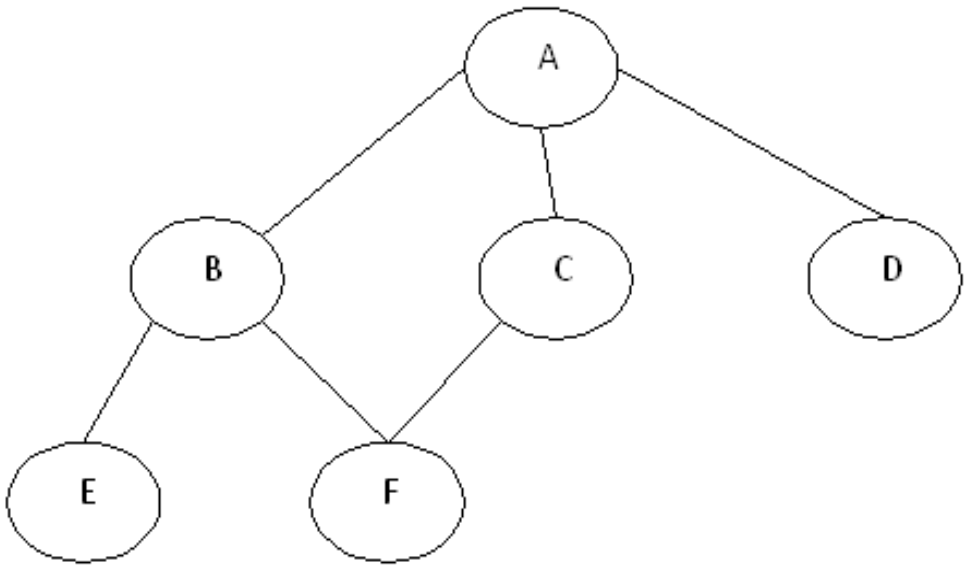
EXP_3 : BFS and DFS

Title : To implement BFS and DFS

Theory :

Breadth First Search (BFS)

This is a very different approach for traversing the graph nodes. The aim of BFS algorithm is to traverse the graph as close as possible to the root node. Queue is used in the implementation of the breadth first search. Let's see how BFS traversal works with respect to the following graph:

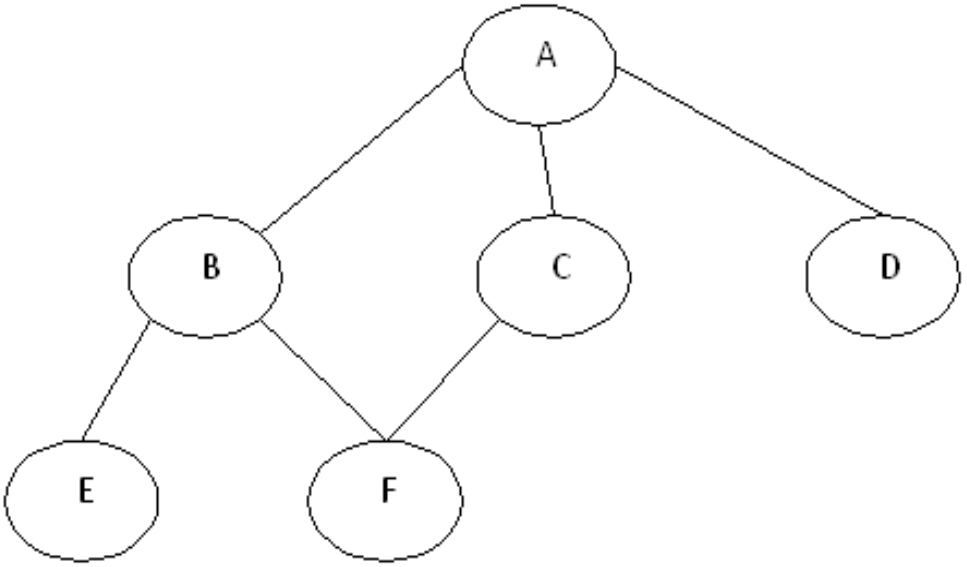


If we do the breadth first traversal of the above graph and print the visited node as the output, it will print the following output. "A B C D E F". The BFS visits the nodes level by level, so it will start with level 0 which is the root node, and then it moves to the next levels which are B, C and D, then the last levels which are E and F.

Depth First Search (DFS)

Depth-first search (**DFS**) is an algorithm for traversing or searching tree or graph data structures. One starts at the root (selecting some arbitrary node as the root in the case of a graph) and explores as far as possible along each branch before backtracking.

The aim of DFS algorithm is to traverse the graph in such a way that it tries to go far from the root node. Stack is used in the implementation of the depth first search. Let's see how depth first search works with respect to the following graph:



As stated before, in DFS, nodes are visited by going through the depth of the tree from the starting node. If we do the depth first traversal of the above graph and print the visited node, it will be "A B E F C D". DFS visits the root node and then its children nodes until it reaches the end node, i.e. E and F nodes, then moves up to the parent nodes.

Difference between BFS and DFS

Parameter	BFS	DFS
Time Complexity	**$O(b^d)$->**if we traversing till the maximum depth. **$O(b^{d+1})$->**next level.	**$O(b^d)$** d-Maximum depth
Space Complexity	**O(bm)** bm->linear space	**O(bm)** bm->linear space
Completeness	**Yes**	**No**
Optimal	**Yes**	**No**

Properties BFS/DFS

- BFS and DFS originating from v create a tree, whose nodes are visited by the algorithm and whose arcs are those traversed. Both create a forest of trees which spans G.
- If w is reachable by DFS and v is visited before, then w is a descendent of v in the DFS forest.
- An arc of a digraph G is a tree arc if it belongs to one of the trees of the forests formed by BFS or DFS.
- DFS may also be used to collect a sample of graph nodes. However, incomplete DFS, similarly to incomplete BFS, is biased towards nodes of high degree.