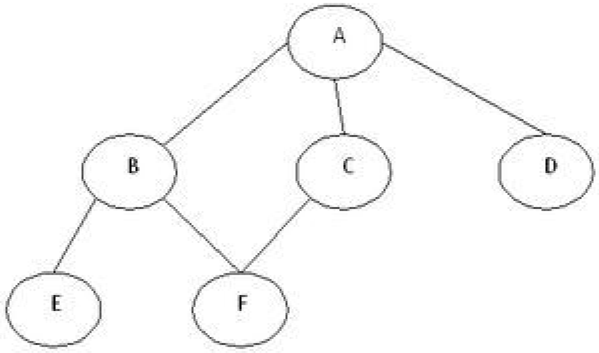
BFS AND DFS

Graph Traversal :

The breadth first search (BFS) and the depth first search (DFS) are the two algorithms used for traversing and searching a node in a graph. They can also be used to find out whether a node is reachable from a given node or not.

1.Depth First Search :

The aim of DFS algorithm is to traverse the graph in such a way that it tries to go as 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 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.

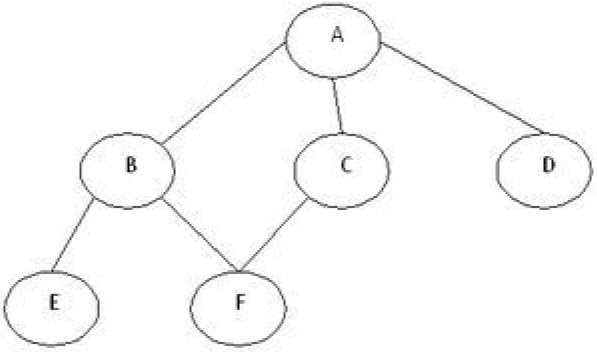
- For further deep understanding , checkout hackerearth tutorial : https://www.hackerearth.com/practice/algorithms/graphs/depth-first-search/tutorial/

Applications :

- Detecting cycle in a graph.
- To find number of connected components of the graph.
- To find all nodes within the same component.
- For finding path between two given vertices (path finding).
  And many others…

2. Breadth First Search :

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.

- For further deep understanding , checkout hackerearth tutorial : https://www.hackerearth.com/practice/algorithms/graphs/breadth-first-search/tutorial/

Applications :

- For finding shortest path in unweighted graph.
- For Finding connected components of the given graph.
- Finding all nodes within the same component.
- For finding path between two given vertices (path finding).
  And many others…

Problems :

Here are the problems,which can be solved by bfs or dfs. They are mixed problems on bfs and dfs.So think yourself which traversal to use for particular question according to need.

- https://www.hackerearth.com/practice/algorithms/graphs/depth-first-search/practice-problems/algorithm/bishu-and-his-girlfriend/
- http://codeforces.com/problemset/problem/500/A
- http://codeforces.com/problemset/problem/902/B
- http://codeforces.com/problemset/problem/3/A (on 2D matrix)
- https://www.hackerearth.com/practice/algorithms/graphs/depth-first-search/practice-problems/algorithm/monk-and-graph-problem/

- http://codeforces.com/problemset/problem/977/E
- http://codeforces.com/problemset/problem/893/C
- https://www.hackerrank.com/contests/ipc-5/challenges/random-1-1/problem
- http://codeforces.com/contest/982/problem/C
- http://codeforces.com/contest/979/problem/C
- http://codeforces.com/problemset/problem/115/A
- http://codeforces.com/problemset/problem/520/B