**Graph Traversal - DFS**

Graph traversal is technique used for searching a vertex in a graph. The graph traversal is also used to decide the order of vertices to be visit in the search process. A graph traversal finds the egdes to be used in the search process without creating loops that means using graph traversal we visit all verticces of graph without getting into looping path.  
  
There are two graph traversal techniques and they are as follows...

1. **DFS (Depth First Search)**
2. **BFS (Breadth First Search)**

**DFS (Depth First Search)**

DFS traversal of a graph, produces a **spanning tree** as final result. **Spanning Tree** is a graph without any loops. We use **Stack data structure** with maximum size of total number of vertices in the graph to implement DFS traversal of a graph.  
  
We use the following steps to implement DFS traversal...

* **Step 1 -**Define a Stack of size total number of vertices in the graph.
* **Step 2 -**Select any vertex as **starting point** for traversal. Visit that vertex and push it on to the Stack.
* **Step 3 -**Visit any one of the **adjacent** vertex of the verex which is at top of the stack which is not visited and push it on to the stack.
* **Step 4 -**Repeat step 3 until there are no new vertex to be visit from the vertex on top of the stack.
* **Step 5 -**When there is no new vertex to be visit then use **back tracking** and pop one vertex from the stack.
* **Step 6 -**Repeat steps 3, 4 and 5 until stack becomes Empty.
* **Step 7 -**When stack becomes Empty, then produce final spanning tree by removing unused edges from the graph

**Back tracking** is coming back to the vertex from which we came to current vertex.

**Example**

