DFS (Depth First Search)

- We traverse the graph in a depth-ward way.
- We use a **stack** so we can remember to get to the next vertex to start searching when we meet a dead end in our traversal.

Logic:

> Visit the adjacent unvisited node.
> Mark it as visited & push it in the stack.
> if (no adjacent node found){
      pop a node from the stack
  } // will pop all nodes from the stack which don't have adjacent nodes
> Repeat process until stack is empty.

```
  > Visit the adjacent unvisited node.
  > Mark it as visited & push it in the stack.
  > if (no adjacent node found){
        pop a node from the stack
    } // will pop all nodes from the stack which don't have adjacent nodes
  > Repeat process until stack is empty.
```

Pseudo Code:

```
DFS(G,s):  // G the graph, s the source node
    // Let S the stack
    S.push(s);
    Mark s as visited  // use a boolean array
     while(S not empty){
          // pop a node from S
          v = S.top();
           // push all neighbors if v into the stack(those who we have not visited!)
           for(all neighbors u of v in G){
               if(u is not visited){
                   S.push(w);
                   Mark w as visited
               }
            }
        }
```