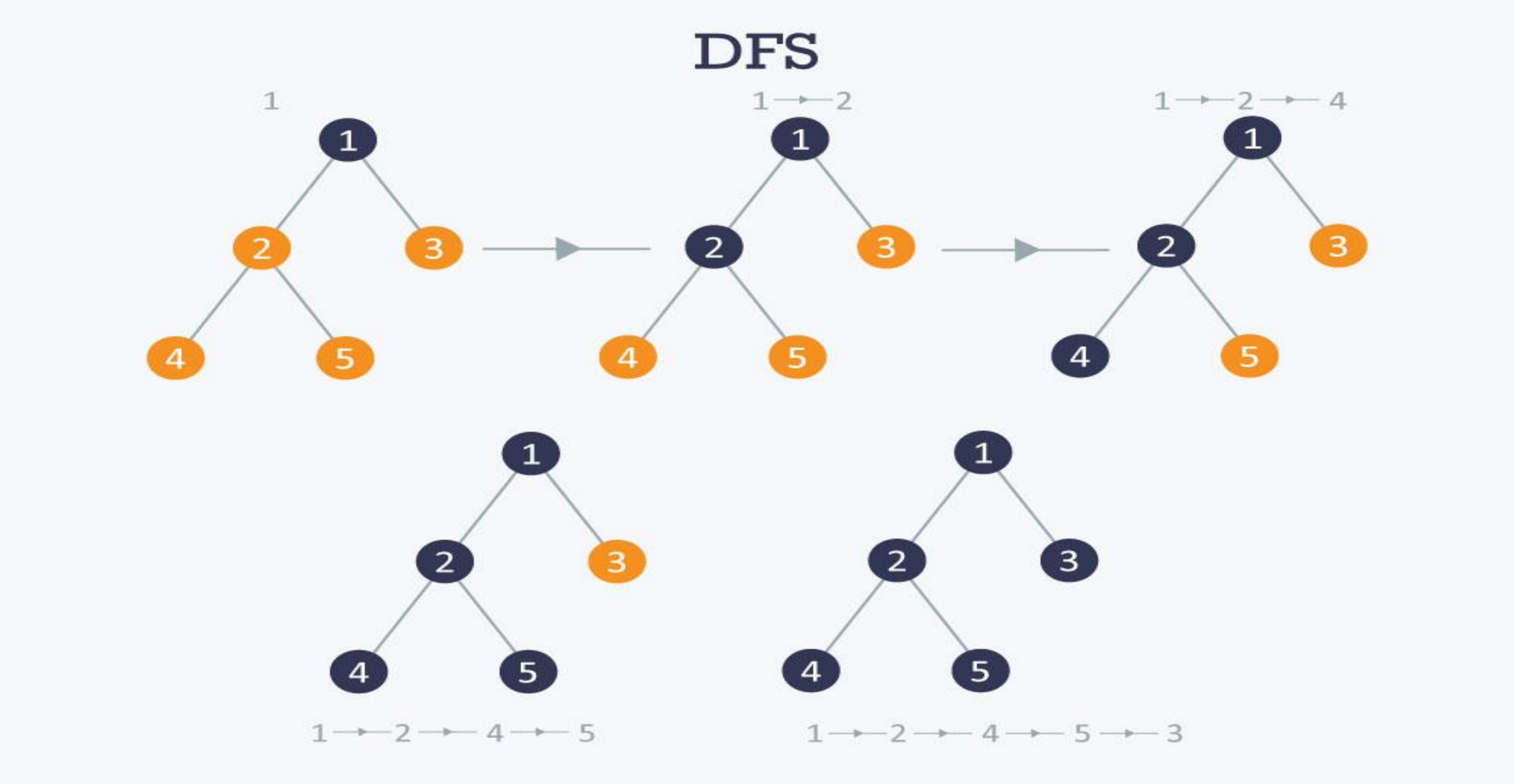
Graph traversal technique II: DFS (Depth First Search)

Presented by: Adarsh Singh

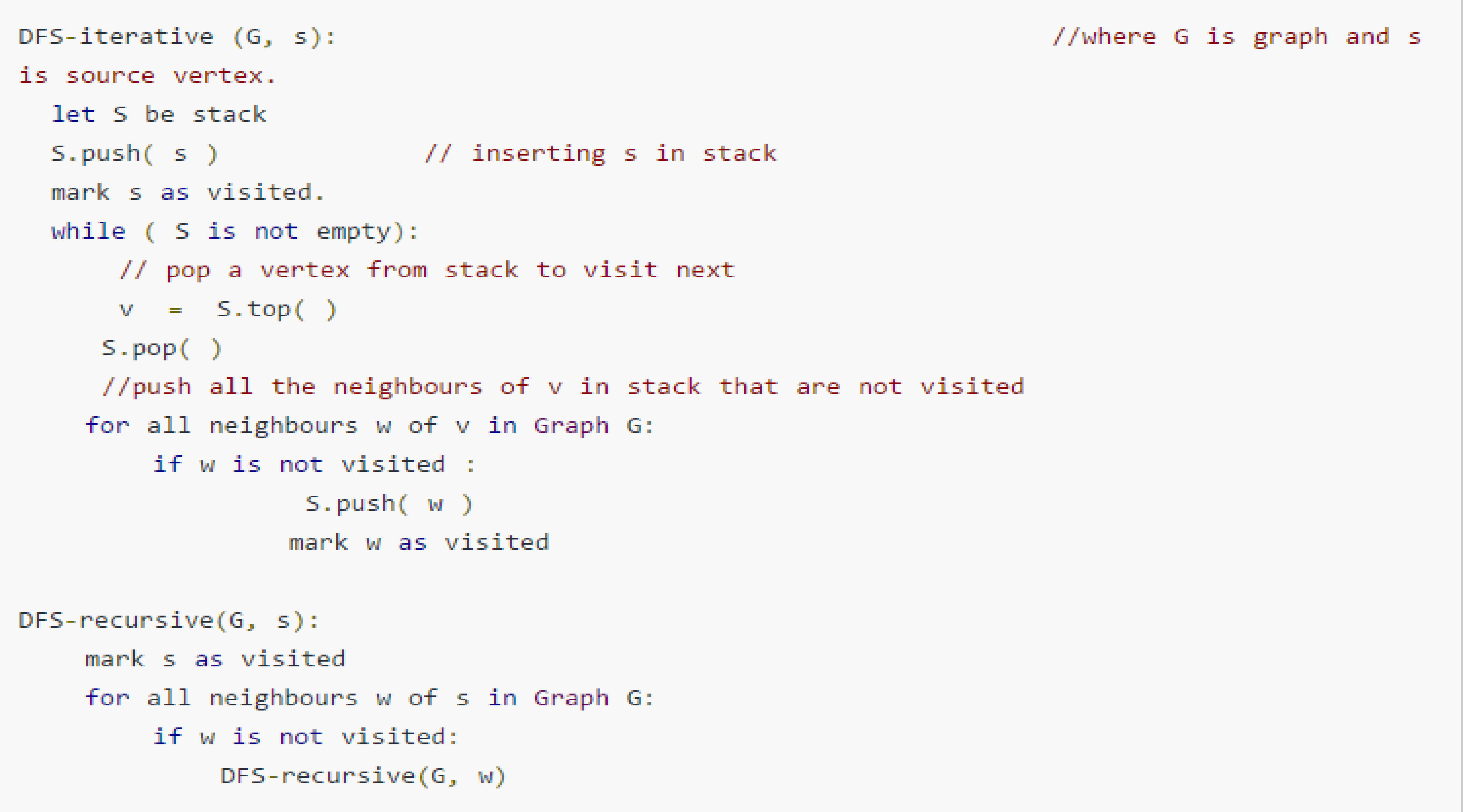
What is DFS?

Depth first search is a recursive algorithm that uses the idea of backtracking. Basically, it involves exhaustive searching of all the nodes by going ahead - if it is possible, otherwise it will backtrack. By backtrack, here we mean that when we do not get any further node in the current path then we move back to the node, from where we can find the further nodes to traverse. In other words, we will continue visiting nodes as soon as we find an unvisited node on the current path and when current path is completely traversed we will select the next path.

Example



Pseudocode



```
DFS-iterative (G, s):                                               //where G is graph and s
is source vertex.
  let S be stack
  S.push( s )               // inserting s in stack
  mark s as visited.
  while ( S is not empty):
       // pop a vertex from stack to visit next
       v  =  S.top( )
      S.pop( )
      //push all the neighbours of v in stack that are not visited
    for all neighbours w of v in Graph G:
         if w is not visited :
                   S.push( w )
                  mark w as visited

DFS-recursive(G, s):
    mark s as visited
    for all neighbours w of s in Graph G:
         if w is not visited:
             DFS-recursive(G, w)
```

Thanks