

# Depth First Search Algorithm in Pseudo-Code

## Recursive (More Natural)

```
// bool visited[vertex] already initialized for all vertices in G
function DFS(vertex src)
{
    if visited[src] {
        return;
    }
    visited[src] = true;
    for each v in src.adj_list() {
        DFS(v);
    }
}
```

## Iterative (Manage Your Own Stack):

- Although this looks awfully similar to BFS, it requires a **stack**, which results in traversal difference.

```
function DFS(vertex src, graph G)
{
    array visited = new();
    for v in G.vertices {
        visited[v] = false;
    }

    Stack S = new();
    S.push(src);
    while !S.empty() {
        vertex u = S.pop();
        if !visited[u] {
            visited[u] = true;
            for each v in u.adj_list() {
                S.push(v);
            }
        }
    }
}
```