Graph Traversal

Depth First Search (DFS)

- Visit all descendants of a node before visiting sibling nodes
- For the implementation we need an array visited[] to keep track of which nodes we've already visited. Its size is the number of nodes.

and the order in which we visited them

- Here's the pseudocode:

```
Set all elements of visited[] to 0
orderNumber = 0

DFS(node) {
    orderNumber ++
    visited[node] = orderNumber
    for each node adjNode adjacent to node
        if (visited[adjNode] == 0)
            DFS(adjNode)
}

for each node node in the graph
    if (visited[node] == 0)
        DFS(node)
```

Uses stack as data structure

Complexity of DFS

- If we represent the graph with an adjacency list the complexity is linear: $O(N+E)$
- If we represent the graph with an adjacency matrix the complexity is quadratic: $O(N^2)$

N: Number of nodes, E: number of edges