

Breadth-First (BFS) and Depth-First Search (DFS)

```
BFS(Graph g, Vertex start)
    create empty "identified" set of vertices
    create empty "visited" set of vertices
    create empty queue of vertices
    create map dist, set all vertices to zero
    create map prev, set all vertices to undefined/null

    identified.add(start)
    queue.enqueue(start)

    while queue is not empty:
        u = queue.dequeue()

        for each vertex v adjacent to u:    // this loop "visits" u
            if v is not in identified:
                identified.add(v)
                queue.enqueue(v)
                dist[v] = dist[u] + 1
                prev[v] = u

        visited.add(u)
```

For Depth-First Search, we can just change Breadth-First Search to use a stack rather than a queue, but a more typical implementation uses recursion:

```
DFS(Graph g, Vertex start)
    create empty "identified" set of vertices
    create empty "visited" set of vertices
    create empty queue of vertices
    DFS-Visit(g, start)

DFS-Visit(Graph g, Vertex u)

    identified.add(u)

    for each vertex v adjacent to u:    // this loop "visits" u
        if v is not in identified:
            DFS-Visit(g, v)

    visited.add(u)
```

