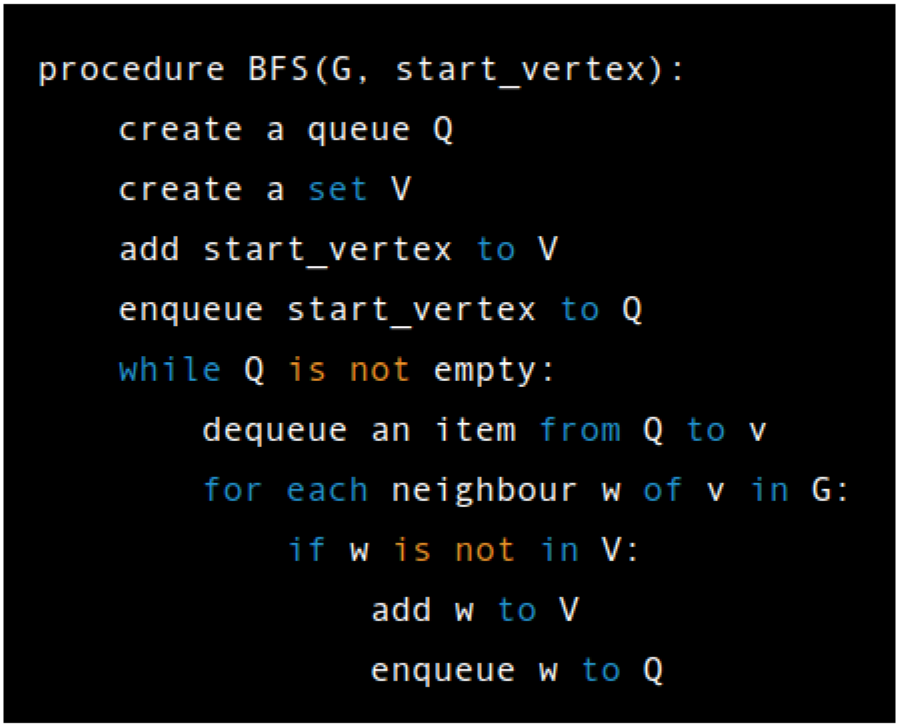
BFS (Breadth-First Search) and DFS (Depth-First Search) are both algorithms for traversing and searching a graph.

BFS, as the name suggests, searches breadth-wise in the graph. It starts at the tree root (or some arbitrary node of a graph) and explores all the neighboring nodes at the present depth prior to moving on to the nodes at the next depth level.

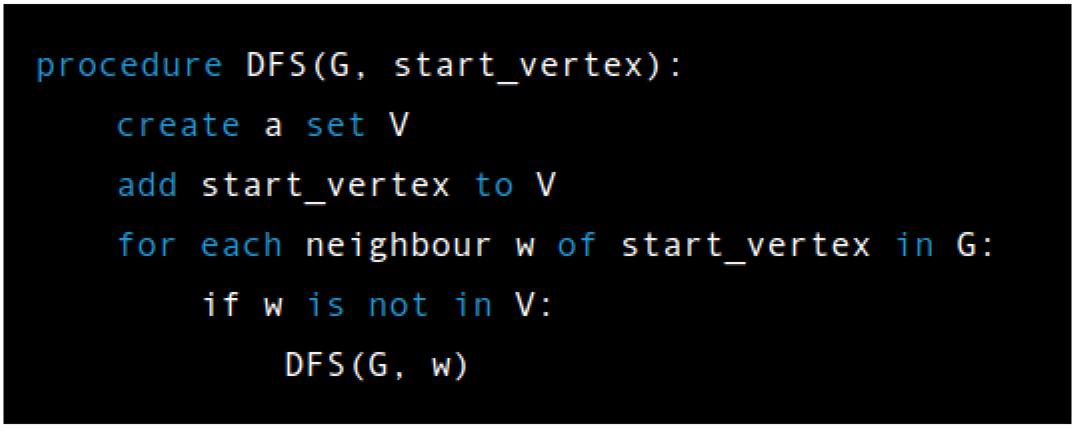
Here's a pseudocode for BFS:



```
procedure BFS(G, start_vertex):
    create a queue Q
    create a set V
    add start_vertex to V
    enqueue start_vertex to Q
    while Q is not empty:
        dequeue an item from Q to v
        for each neighbour w of v in G:
            if w is not in V:
                add w to V
                enqueue w to Q
```

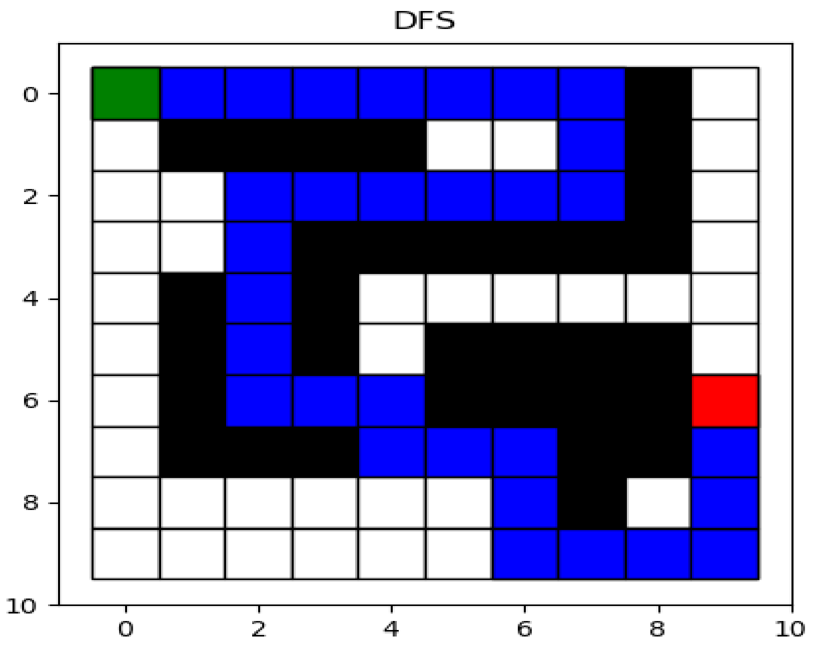
DFS, on the other hand, searches depth-wise in the graph. It starts at the root node and explores as far as possible along each branch before backtracking.

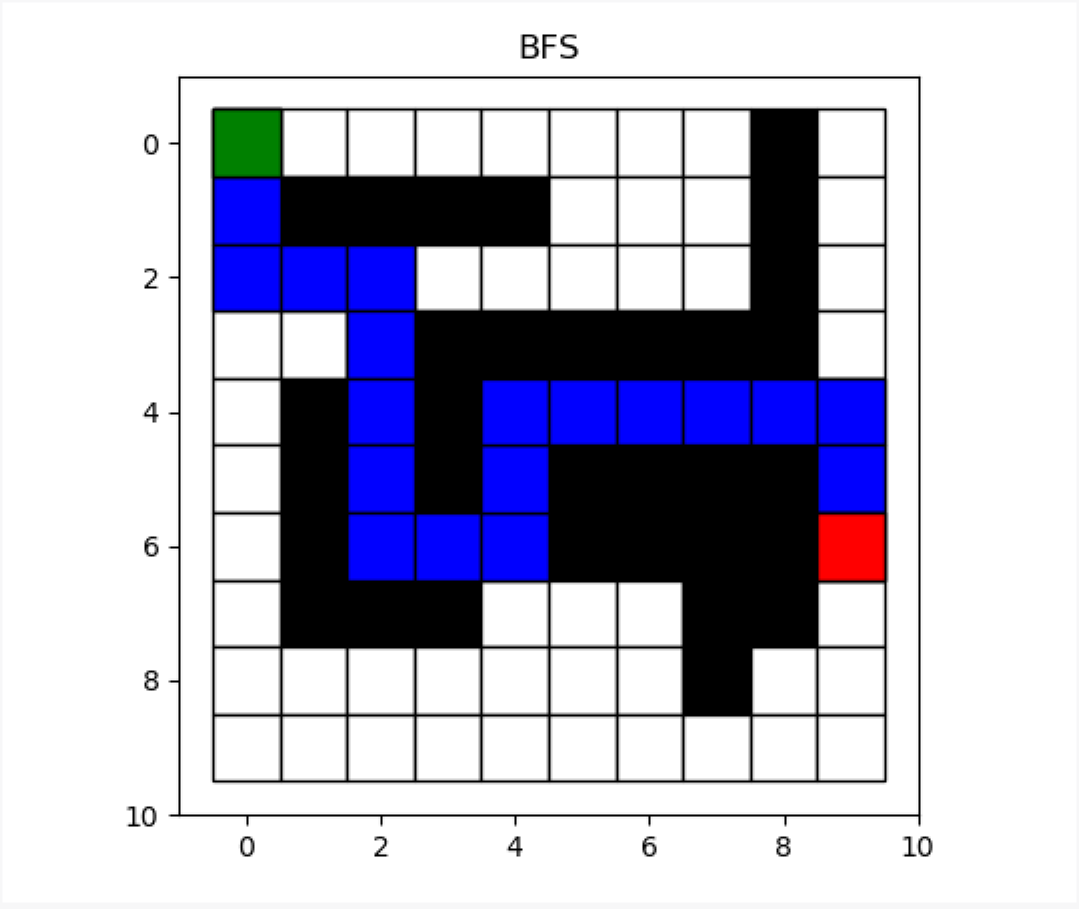
Here's a pseudocode for DFS:



```
procedure DFS(G, start_vertex):
    create a set V
    add start_vertex to V
    for each neighbour w of start_vertex in G:
        if w is not in V:
            DFS(G, w)
```

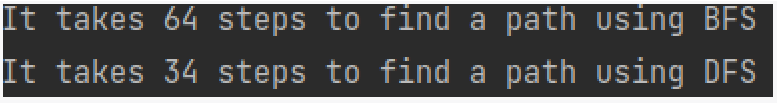
In summary, BFS uses a queue data structure while DFS uses a stack data structure. BFS visits all vertices at a distance k before visiting vertices at a distance k+1 while DFS visits a vertex as soon as it is discovered. Also, BFS is used when the shortest path is to be determined from one node to another node. DFS is used when the solution is required to go deep in any particular branch before exploring others.





Here we see that DFS and BFS paths are different. This is because BFS will explore all the neighbors first and then go deep whereas DFS will first go deep into one of the neighbors before exploring other neighbors.

Because both algorithms have a different approach to search the number of steps taken is different



```
It takes 64 steps to find a path using BFS
It takes 34 steps to find a path using DFS
```

References:

https://www.geeksforgeeks.org/depth-first-traversal-dfs-on-a-2d-array/
Lecture slides