DFS

Diego Araque - A01026037
Marco Torres - A01025334
Fernando Valdeón - A01745186
Uriel Aguilar - A01781698

29 de Septiembre del 2022

DFS or Depth First Search is a computational algorithm that traverses a graph, this means that the algorithm will visit each and all of the nodes that are present in a certain order. It takes an undirected or directed graph $G = (V, E)$, a set of vertex and edges.

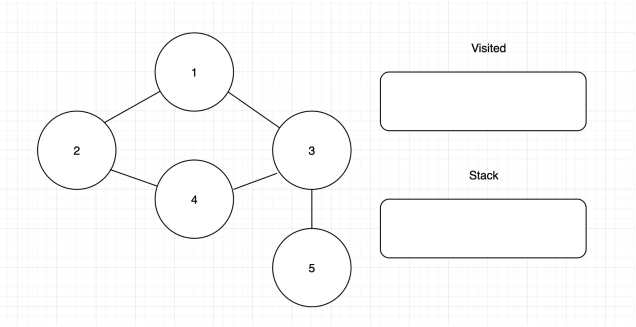
The algorithm starts at the root node and explores as far as possible along each branch before backtracking. Take into account that the root node can be any node of the graph, and it will still traverse all of it.

To implement this algorithm we will use a stack, which means that the last nodes to be pushed into it will be the first ones out. Different from the BFS (where a queue is used), where the first node in is the first node out. The steps of the algorithm we are going to implement are the following:

1.- First of all the program will read a .txt file to create the graph
2.- Then an empty stack and an empty list that will contain the visited nodes is created
3.- The root node is going to be added to the visited list, and all of its adjacent nodes will be added to the stack.
4.- Now the first node of the stack will be added to the visited list, and all of its adjacent nodes that are univisted will be added to the stack. This process will be done several times, until the stack is empty, which means that all of the nodes will be visited.

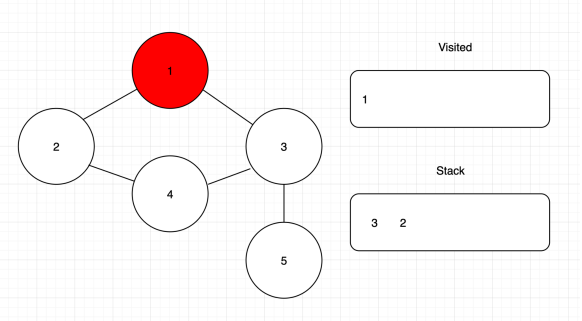
In the following graph we will explain how the algorithm works:

Iteration #0



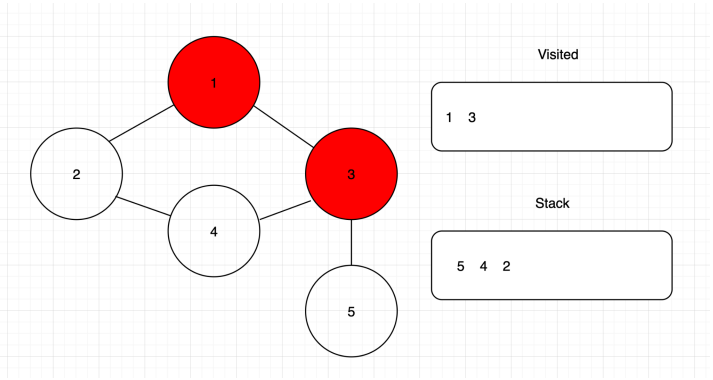
In this iteration we just show the base graph, which will explain the process done to traverse all of it with dfs.

Iteration #1



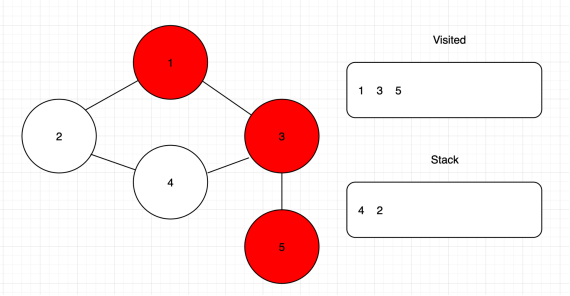
Next, we take the element on the top of the stack, mark as visited and add their adjacent nodes to the stack.

Iteration #2



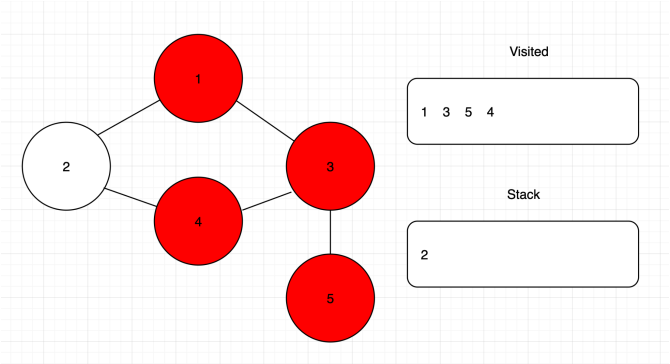
We now introduce the next element of stack to the visited nodes and check if it has any adjacent nodes, and add the ones that are not visited to the stack. In this case, node 3 has 2 adjacent nodes and all of them are unvisited. Therefore nodes 4 and 5 are added to the stack.

Iteration #3



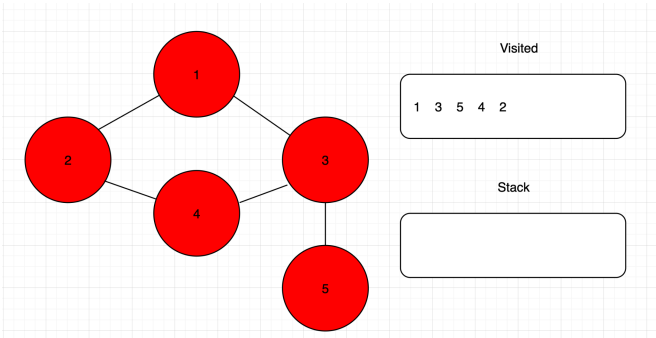
In this iteration we check insert the first node of the stack, in this case node 5 is added to the stack. We check for its unvisited adjacent nodes, since it has none, the stack is not updated.

Iteration #4



In this iteration we add node 4 to the visited nodes, and check for its univisited adjacent nodes. Node 2 is univisited, but since it is already in the stack we don't add it.

Iteration #5



For this graph this will be the last iteration, and we just add the only element left on the stack, which is node 2. Since the stack is now empty, the iteration stops and we have traversed all of the graph.

Referencias:

Depth First Search (DFS) Algorithm. (2022). Retrieved September 26, 2022, from Programiz.com website: https://www.programiz.com/dsa/graph-dfs