A DEPTH-FIRST SEARCH SIMULATION

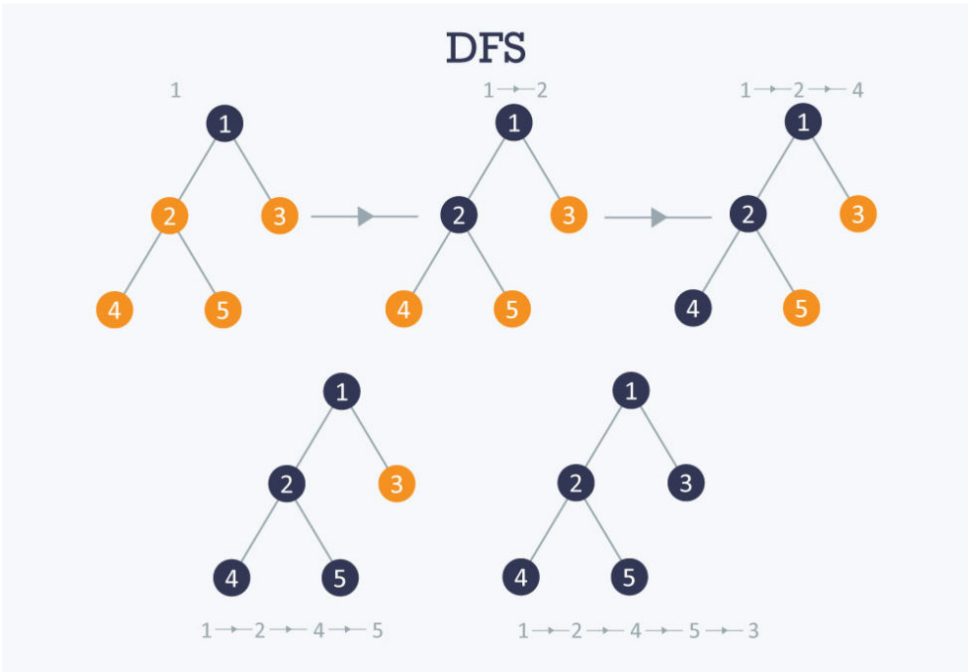
Submitted to:
Dr. Norshuhani Zamin

Submitted by:
CAI, EDISON
CHEN, TING RUNG
DIONISIO, KARL ALFONSO
ESTEBAL, EIDRENE GLENA

S13

I. Introduction

Depth-First Search (DFS) Algorithm is an algorithm for searching through tree or graph data structures. The algorithm begins at the root node and proceeds to traverse each branch of the root node to its full vertical length, as far as it can go, before backtracking and proceeding from left to right (*Depth First Search or DFS for a Graph*, 2023). In the context of DFS, backtracking is going back to the parent node when a node with all its adjacent/child nodes has been visited and it has no sibling nodes (*Print the DFS Traversal Step-Wise (Backtracking Also)*, 2023). DFS can be executed through recursive calls, wherein the recursive calls generate a call stack that maintains a record of the nodes awaiting visitation or the nodes already visited (Sryheni, 2023).



DFS diagram

In this diagram (*Water Jug Problem Using DFS*, 2010), the nodes are traversed as deep as possible along each branch before backtracking and then proceeding from left to right. In traversing the whole graph using DFS, the path that came out of it is from node 1 to node 3. A stepwise approach would look like this:

1. node 1 -> node 2
2. node 1 -> node 2 -> node 4
3. node 1 -> node 2 -> node 4 -> node 5
4. node 1 -> node 2 -> node 4 -> node 5 -> node 3

Note that since node 5 has no other sibling nodes that have not been visited, traversal of the graph backtracked and went to node 3.

DFS requires less memory since only the nodes on the current path are stored and that they may find a solution without examining much of the space at all (*Depth-First Search Example Advantages and Disadvantages*, n.d.). On the other hand, the disadvantage of DFS is that there is a possibility that it may go down the left-most path the whole time and not find the most minimal solution, if there is more than one solution (*Applications, Advantages and Disadvantages of Depth First Search (DFS)*, 2023).

DFS is commonly used for solving problems that only have one solution (e.g. mazes and sudoku puzzles), web crawling (exploring the links on a website), model checking (the process of checking if a system model meets its set criteria), and cycle detections and route mapping (*Applications, Advantages and Disadvantages of Depth First Search (DFS)*, 2023; Janghu, 2022). It is also worth remembering that DFS does not always guarantee finding the shortest path between two nodes, as it focuses on exploring deeper into the graph rather than exploring it laterally (*Shortest Path: DFS, BFS or Both?*, 2013).

II. Pseudocode

This is the process that showed how the depth-first search algorithm operates (Srivastava, 2023):

1. Begin at a node in the graph or tree.
2. Visit the current node and make it as visited.
3. Explore neighboring/child nodes that have not been visited yet.
4. Select one of the unvisited neighbors and recursively execute steps 2 and 3 for it.
5. If there are no unvisited neighbors, return/backtrack to the previous node and repeat step 4 for any unvisited neighbors of that node.
6. Repeat steps 4 and 5 until all nodes have been visited.

<u>Pseudocode (Landup et al., n.d.):</u>

```
DFS(graph, node):
    node.visited = true
    for each neighboring_nodes in graph.adjacent[node]:
        if !(neighboring_nodes.visited):
        DFS(graph, neighboring_nodes)
```

III. Results and Analysis

A. Strengths and Limitations

This program provides a variety of choices right from the start and every time another choice is made. This provides a more user-friendly interface that is fit for viewing the current graph and the path the DFS will traverse. However, this program did not include the weight of the path it will take. If ever there will be, this Depth-First Search Algorithm Simulation would have a more accurate way of depicting the path traveled from the start node to the goal node.

B. Results

1. Running the program

```
G:\My Drive\school\AY 2022-2023\term 3\csintsy\mco\mco1>python dfs_updated.py
[a] Add node
[b] Delete node
[c] Edit node
[d] See current graph
[e] DFS Search
[f] Exit the program

Enter your option:
```

2. Adding a node

```
[a] Add node
[b] Delete node
[c] Edit node
[d] See current graph
[e] DFS Search
[f] Exit the program

Enter your option: a

Enter parent node: 0
Enter child node: 1
Enter the position to insert the child node (optional): 1

[a] Add node
[b] Delete node
[c] Edit node
[d] See current graph
[e] DFS Search
[f] Exit the program

Enter your option: d

Current Graph:
0 -> 1
```

3. Deleting a node

```
[a] Add node
[b] Delete node
[c] Edit node
[d] See current graph
[e] DFS Search
[f] Exit the program

Enter your option: a

Enter parent node: 0
Enter child node: 2
Enter the position to insert the child node (optional): 2

[a] Add node
[b] Delete node
[c] Edit node
[d] See current graph
[e] DFS Search
[f] Exit the program

Enter your option: d

Current Graph:
0 -> 1, 2
```

```
[a] Add node
[b] Delete node
[c] Edit node
[d] See current graph
[e] DFS Search
[f] Exit the program

Enter your option: b

Enter node to delete: 2

[a] Add node
[b] Delete node
[c] Edit node
[d] See current graph
[e] DFS Search
[f] Exit the program

Enter your option: d

Current Graph:
0 -> 1
```

4. **Editing a node**

```
Current Graph:
0 -> 1, 2

[a] Add node
[b] Delete node
[c] Edit node
[d] See current graph
[e] DFS Search
[f] Exit the program

Enter your option: c

Enter node to edit: 2
Enter new node value: 456

[a] Add node
[b] Delete node
[c] Edit node
[d] See current graph
[e] DFS Search
[f] Exit the program

Enter your option: d

Current Graph:
0 -> 1, 456
```

5. **Presenting the current graph**

```
[a] Add node
[b] Delete node
[c] Edit node
[d] See current graph
[e] DFS Search
[f] Exit the program

Enter your option: a

Enter parent node: 0
Enter child node: 789
Enter the position to insert the child node (optional):

[a] Add node
[b] Delete node
[c] Edit node
[d] See current graph
[e] DFS Search
[f] Exit the program

Enter your option: d

Current Graph:
0 -> 1, 456, 789
```

6. Performing the DFS search

```
Current Graph:
0 -> 1, 2, 3
1 -> 4, 5, 6
2 -> 7, 8, 9

[a] Add node
[b] Delete node
[c] Edit node
[d] See current graph
[e] DFS Search
[f] Exit the program

Enter your option: e

Enter the start node for DFS traversal: 0

Enter the goal node for DFS traversal: 8

Following is the Depth-first search:
0 -> 1 -> 4 -> 5 -> 6 -> 2 -> 7 -> 8

Found the goal node "8"
```

7. Exiting the program

```
Current Graph:
0 -> 1, 2, 3
1 -> 4, 5, 6
2 -> 7, 8, 9

[a] Add node
[b] Delete node
[c] Edit node
[d] See current graph
[e] DFS Search
[f] Exit the program

Enter your option: e

Enter the start node for DFS traversal: 0

Enter the goal node for DFS traversal: 8

Following is the Depth-first search:
0 -> 1 -> 4 -> 5 -> 6 -> 2 -> 7 -> 8

Found the goal node "8"

[a] Add node
[b] Delete node
[c] Edit node
[d] See current graph
[e] DFS Search
[f] Exit the program

Enter your option: f
Terminating code. Goodbye!
```

References

Applications, Advantages and Disadvantages of Depth First Search (DFS). (2023, May 31). GeeksforGeeks. Retrieved June 27, 2023, from https://www.geeksforgeeks.org/applications-of-depth-first-search/

Depth-first search Example Advantages and Disadvantages. (n.d.). VTUPulse. Retrieved June 27, 2023, from https://www.vtupulse.com/artificial-intelligence/depth-first-search-example-advantages-and-disadvantages/

Depth First Search or DFS for a Graph. (2023, June 9). GeeksforGeeks. Retrieved June 27, 2023, from https://www.geeksforgeeks.org/depth-first-search-or-dfs-for-a-graph/

Janghu, R. (2022, December 13). Depth First Search (DFS) in Python. Scaler. Retrieved June 28, 2023, from https://www.scaler.com/topics/dfs-python/

Landup, D., Niketić, A., Stamenic, D., & Ninkovic, J. (n.d.). *Graphs in Python - Theory and Implementation - Depth-First Search (DFS) Algorithm*. Stack Abuse. Retrieved June 29, 2023, from https://stackabuse.com/courses/graphs-in-python-theory-and-implementation/lessons/depth-first-search-dfs-algorithm/

Print the DFS traversal step-wise (Backtracking also). (2023, January 27). GeeksforGeeks. Retrieved June 27, 2023, from https://www.geeksforgeeks.org/print-the-dfs-traversal-step-wise-backtracking-also/

Shortest path: DFS, BFS or both? (2013, February 9). Stack Overflow. Retrieved June 28, 2023, from https://stackoverflow.com/questions/14784753/shortest-path-dfs-bfs-or-both

Srivastava, A. (2023, March 4). *DFS (Depth First Search) Algorithm*. codingninjas. Retrieved June 29, 2023, from https://www.codingninjas.com/studio/library/dfs-depth-first-search-algorithm

Sryheni, S. (2023, March 24). *Introduction to Depth First Search Algorithm (DFS)*. Baeldung. Retrieved June 29, 2023, from https://www.baeldung.com/cs/depth-first-search-intro

Water Jug Problem using DFS. (2010, April 7). *DFS diagram* [Image]. Kartheek Chava. Retrieved June 28, 2023, from https://kchava.com/computer-science/water-jug-problem-using-dfs/

IV. Contributions of each Member

Group Member	Source Code	Report
CAI, EDISON		
CHEN, TING RUNG	- Fix the bug of the code	- Results and Analysis
DIONISIO, KARL ALFONSO	- Added the goal node functionality, edit existing nodes, and overall main menu of the code as well as a readme.txt file for instructions	
ESTEBAL, EIDRENE GLENA	- Started the source code with the functions dfs(), visited(), add_child(), and print_current_graph()	- Did the Introduction, Pseudocode, and Results and Analysis