# California State University, Fresno

# DEPARTMENT OF COMPUTER SCIENCE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class: | **Algorithms & Data Structures** | | | Semester: | **Fall 2021** |
|  | | | | | |
| Points |  | Document author: | **Saishnu Ramesh Kumar** | | |
|  | Author’s email: | **saishnu\_rk@mail.fresnostate.edu** | | |
| Laboratory number: | **Section 1, 11am to 12:50pm** | | |
|  | | | | | |

**1. Statement of Objectives**

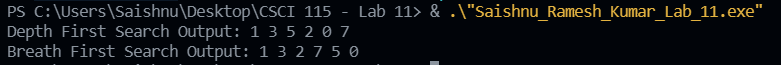
This lab assignment deals with using Depth First Search and Breadth First Search. Depth First Search works where the program would traverse from the top of the node down to the bottom of the tree. This method would keep repeating that until it completely goes through the entire tree. Breadth First Search works in the sense where it traverses for each level until the entire tree is gone through. Also, Breath First Search works by using the queue data structure. The time complexity of both Depth and Breadth First Search is O (V + E) where V is the number of vertices in the graph and E is the number of edges within the graph.

**2. Experimental Procedure**

For this assignment, I created two classes respectively, a DepthFirstSearch and a BreadthFirstSearch class and included all their respective functions within themselves. A vertex and an adjacency list was declared within both classes. The addEdge functions in both classes basically do the same thing as they were assigned to add v into the list of u. The DFS Function contained the part where it will mark the respective vertices as not visited if it has not been visited and would call the DFS\_Util function to traverse through the graph recursively. The BFS on the other hand would also mark the respective vertices as not visited first and it would then, using the queue data structure, start to push/enqueue the node. After that it would pop/dequeue the vertex from the queue and if anything is not visited yet, it would push the adjacent into the queue again and repeat. The classes are then called in the main function with the inputs entered in the main function as well.

**3. Analysis**

Terminal Screenshot Output:

****

**4. Encountered Problems**

I came across a couple of compiler errors, but I was able to fix them and get the program compiling and running again.

**5. Conclusions**

This concept is also fairly new to me but after completing the lab assignment and attending the lab, I believe that I have got a better understanding about how both of these concept’s work. It is interesting to learn how both first search methods work and how one takes a longer time to execute as compared to the other.

**6. References**

Slides provided by TA.

Slides and lecture videos provided by instructor.

<https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/>

<https://www.geeksforgeeks.org/depth-first-search-or-dfs-for-a-graph/>

<https://www.tutorialspoint.com/data_structures_algorithms/depth_first_traversal.htm>

<https://dev.to/danimal92/difference-between-depth-first-search-and-breadth-first-search-6om>