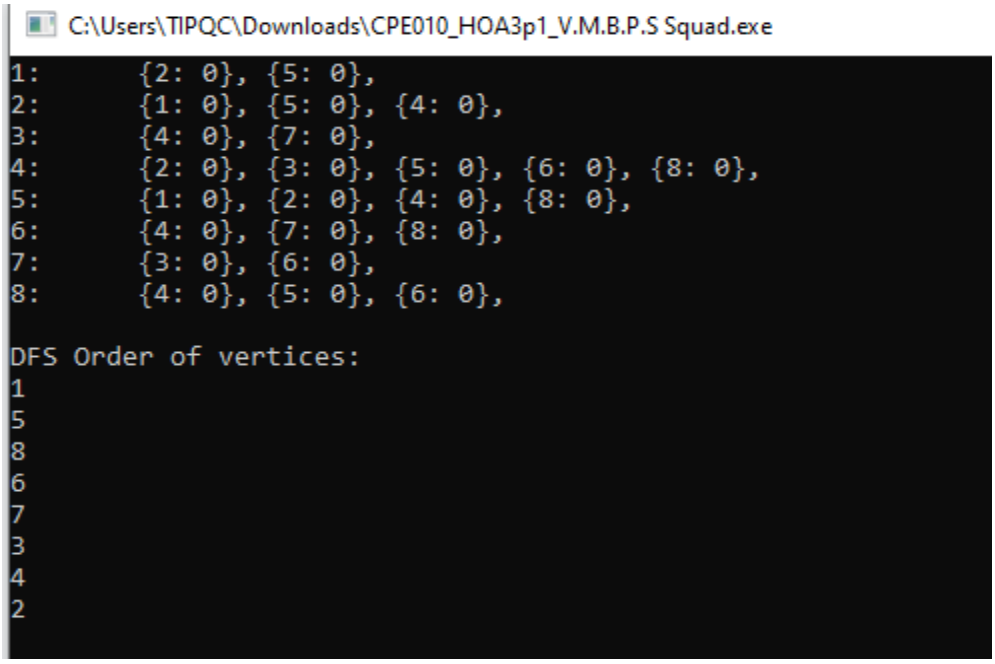
Activity No. 12	
Implementing Depth First Search	
Course Code: CPE010	**Program:** Computer Engineering
Course Title: Data Structures and Algorithms	**Date Performed:** 11/27/2024
Section: CPE21S1	**Date Submitted:** 11/27/2024
Name(s): Abarabar, John Nathan R. Aduana, Jake Norwin Bulambao, Adrian Justin M. Gaspar, Aaron Rowen Potestades, North Nygel G.	**Instructor:** Engr. Sayo

6. Output



7. Supplementary Activity

Answer the following questions:

1. A person wants to visit different locations indicated on a map. He starts from one location (vertex) and wants to visit every vertex until it finishes from one vertex, backtracks, and then explore other vertex from same vertex. Discuss which algorithm would be most helpful to accomplish this task.

- The algorithm that would be most useful for this task would be the Depth First Search, as the flow of the person's needs is similar to the flow that a DFS is performed. They travel from the first location and visit every location possible in one path, and then it backtracks to the first location with another path to explore. It repeats this process until each vertex is labeled as visited, following the same flow as how the person wants to visit different locations.

2. Describe a situation where in the DFS of a graph would possibly be unique.

- Depth First Search or DFS is unique depending on the graph's structure overall. We can see this as an example in a directed graph with a lot of subgraphs that are disconnected where the DFS will only explore one subgraph that will start at a certain vertex. Vertices can be in a unique order depending on the graph's structure and the starting vertex. The DFS of a graph can produce a unique order based on its structure and the order of edges being traversed where if the conditions are met it can produce a unique graph.

3. Demonstrate the maximum number of times a vertex can be visited in the DFS. Prove your claim through code and demonstrated output.

C/C++

```
#include <bits/stdc++.h>
using namespace std;

void addEdge(vector<vector<int>> &adj, int s, int t){
    adj[s].push_back(t);
    adj[t].push_back(s);
}

// Recursive function for DFS traversal
void DFSRec(vector<vector<int>> &adj, vector<bool> &visited,int s){
    // Mark the current vertex as visited
    visited[s] = true;

    // Print the current vertex
    cout << s << " ";

    // Recursively visit all adjacent vertices that are not visited yet
    for (int i : adj[s])
        if (visited[i] == false)
            DFSRec(adj, visited, i);
}

// Main DFS function to perform DFS for the entire graph
void DFS(vector<vector<int>> &adj){
    vector<bool> visited(adj.size(), false);

    // Loop through all vertices to handle disconnected graph
    for (int i = 0; i < adj.size(); i++){
        if (visited[i] == false){
            // If vertex i has not been visited,
            // perform DFS from it
            DFSRec(adj, visited, i);
        }
    }
}

int main(){
    int V = 6;
    // Create an adjacency list for the graph
    vector<vector<int>> adj(V);

    // Define the edges of the graph
    vector<vector<int>> edges = {{1, 2}, {2, 0}, {0, 3}, {4, 5}};

    // Populate the adjacency list with edges
    for (auto &e : edges)
        addEdge(adj, e[0], e[1]);

    cout << "Complete DFS of the graph:" << endl;
    DFS(adj);

    return 0;
}
```

```
Complete DFS of the graph:
0 2 1 3 4 5
```

- The maximum number of times a vertex can be visited in a DFS depends on if the graph is directed or undirected, with the maximum being one in a directed graph, and two in an undirected graph. We can see this being demonstrated in the flow of the code shown above, and the output shown.

4. What are the possible applications of the DFS?

- There are many possible applications where Depth First Search can be used, an example of this would be in path finding for a GPS. Another application of the DFS would be to solve a puzzle which has only one possible solution, like a maze or sudoku puzzle.

5. Identify the equivalent of DFS in traversal strategies for trees. In order to efficiently answer this question, provide a graphical comparison, examine pseudocode and code implementation.

- The equivalent of DFS for a tree would be a post-order traversal, as they both start from the bottom and make their way up their respective data structures.

8. Conclusion

In conclusion, graphs are a data structure which can be observed quite a lot in real life, in things like roads and flight maps, and can be utilized in many ways in code. The procedure this time around was quite easy, needing to just follow the steps given to us by the manual. The supplementary activity was a little challenging this time as well, as we needed to use code to prove our answer to one of the questions, specifically number 3. I think we did quite well in this activity, and if there is anything to improve, I would say that it would just be speed again.

9. Assessment Rubric

Rubric for SO 7 (3)

Criteria	Ratings						Pts
SO 7 PI 1 IILO4 Utilize lifelong learning skills in pursuit of personal development and excellence in professional practice. threshold: 4.8 pts	6 pts Excellent | Educational interests and pursuits exist and flourish outside classroom requirements,knowledge and/or experiences are pursued independently and applies knowledge learned into practice	5 pts Good | Educational interests and pursuits exist and flourish outside classroom requirements,knowledge and/or experiences are pursued independently	4 pts Satisfactory | Look beyond classroom requirements, showing interest in pursuing knowledge independently	3 pts Unsatisfactory | Begins to look beyond classroom requirements, showing interest in pursuing knowledge independently	2 pts Poor | Relies on classroom instruction only	1 pts Very Poor | No initiative or interest in acquiring new knowledge	6 pts
SO 7 PI 2 IILO4 Utilize lifelong learning skills in pursuit of personal development and excellence in professional practice. threshold: 4.8 pts	6 pts Excellent | Completes an assigned task independently and practices continuous improvement	5 pts Good | Completes an assigned task without supervision or guidance	4 pts Satisfactory | Requires minimal guidance to complete an assigned task	3 pts Unsatisfactory | Requires detailed or step-by-step instructions to complete a task	2 pts Poor | Shows little interest to complete a task independently	1 pts Very Poor | No interest to complete a task independently	6 pts
SO 7 PI 3 IILO4 Utilize lifelong learning skills in pursuit of personal development and excellence in professional practice. threshold: 4.8 pts	6 pts Excellent | Synthesizes and integrates information from a variety of sources; formulates a clear and precise perspective; draws appropriate conclusions	5 pts Good | Evaluate information from a variety of sources; formulates a clear and precise perspective.	4 pts Satisfactory | Analyze information from a variety of sources; formulates a clear and precise perspective.	3 pts Unsatisfactory | Apply the gathered information to formulate the problem	2 pts Poor | Gather and summarized the information from a variety of sources but failed to formulate the problem	1 pts Very Poor | Gather information from a variety of sources	6 pts
SO 7 PI 4 IILO4 Utilize lifelong learning skills in pursuit of personal development and excellence in professional practice. threshold: 4.8 pts	6 pts Excellent | Ideas are combined in original and creative ways in line with the new and emerging technology trends to solve a problem or address an issue.	5 pts Good | Ideas are creative and adapt the new knowledge to solve a problem or address an issue	4 pts Satisfactory | Ideas are creative in solving a problem, or address an issue	3 pts Unsatisfactory | Shows some creative ways to solve the problem	2 pts Poor | Shows initiative and attempt to develop creative ideas to solve the problem	1 pts Very Poor | Ideas are copied or restated from the sources consulted	6 pts
							Total Points: 24