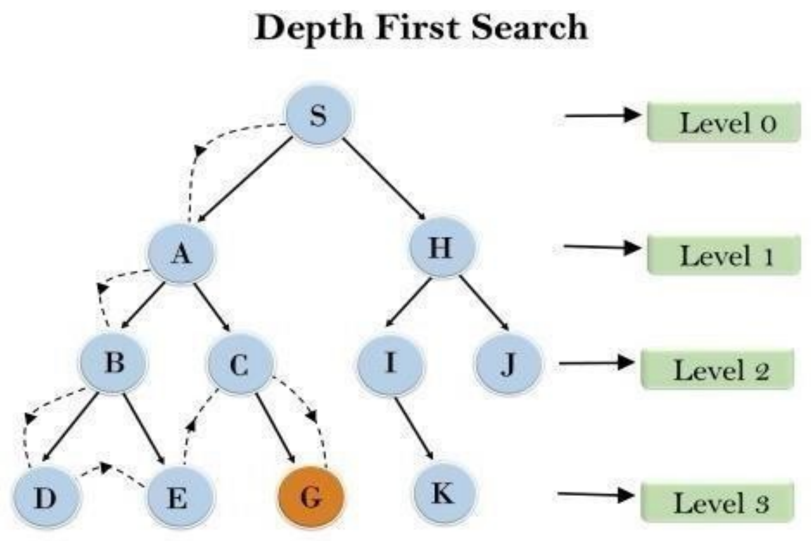
EX.NO:4 **DATE:4/9/2024**
Reg.no:220701013

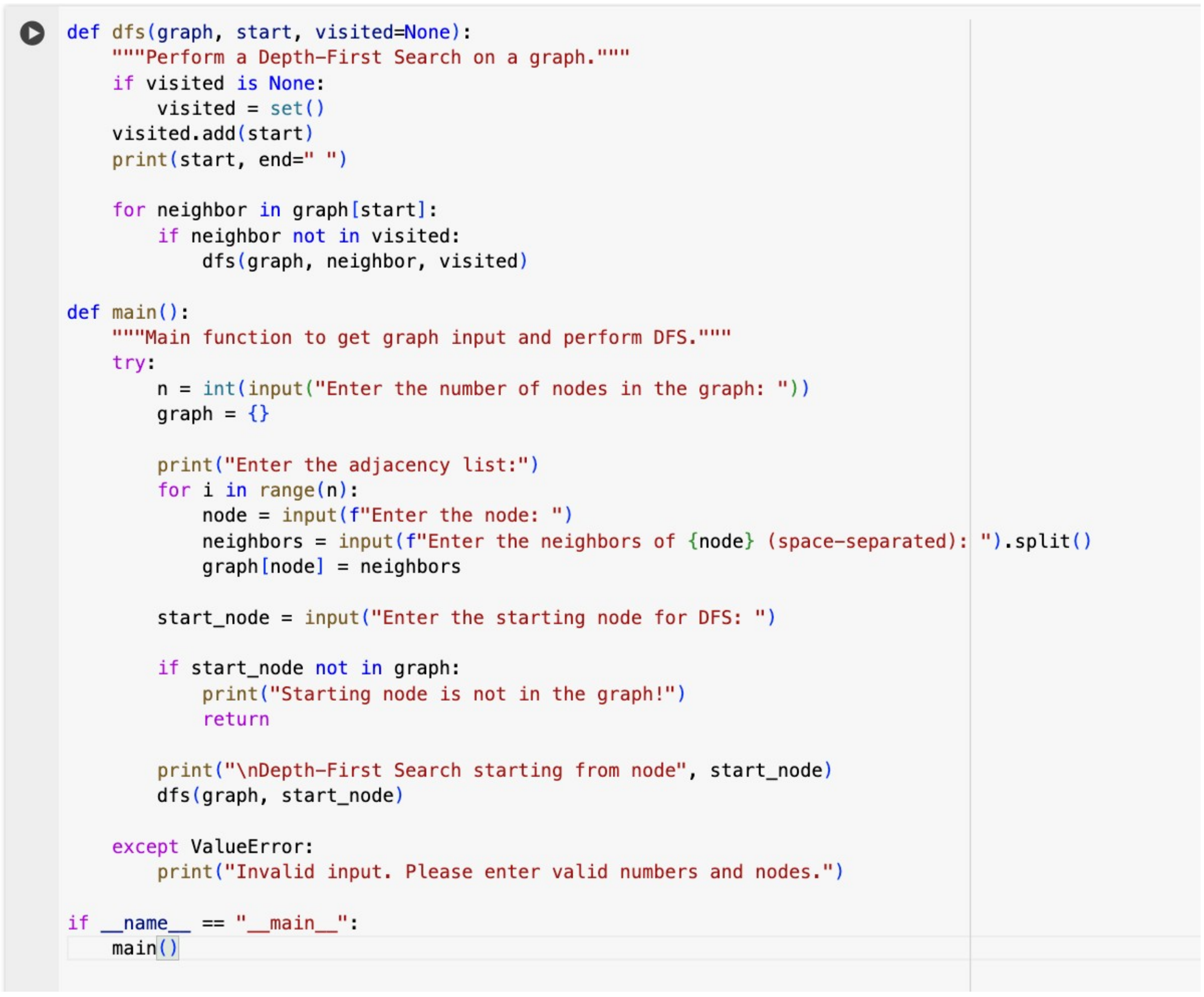
DEPTH-FIRST SEARCH

AIM:
To implement a depth-first search problem using Python

- Depth-first search (DFS) algorithm or searching technique startswith the root node of graph G, and then travel deeper and deeper until we find the goal node or the node which has no children byvisiting different node of the tree.
- The algorithm, then backtracks or returns back from the dead endor last node towards the most recent node that is yet to becompletely unexplored.
- The data structure (DS) which is being used in DFS Depth-firstsearch is stack. The process is quite similar to the BFS algorithm.
- In DFS, the edges that go to an unvisited node are called discovery edges while the edges that go to an already visited nodeare called block edges.

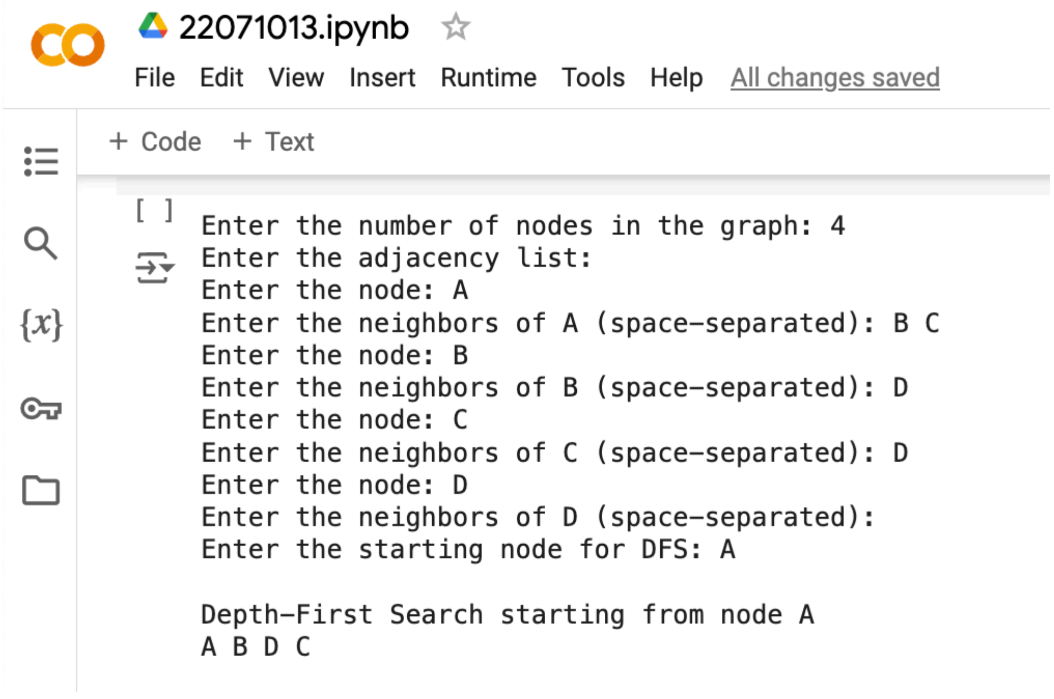


CODE:



```python
def dfs(graph, start, visited=None):
    """Perform a Depth-First Search on a graph."""
    if visited is None:
        visited = set()
    visited.add(start)
    print(start, end=" ")

    for neighbor in graph[start]:
        if neighbor not in visited:
            dfs(graph, neighbor, visited)

def main():
    """Main function to get graph input and perform DFS."""
    try:
        n = int(input("Enter the number of nodes in the graph: "))
        graph = {}

        print("Enter the adjacency list:")
        for i in range(n):
            node = input(f"Enter the node: ")
            neighbors = input(f"Enter the neighbors of {node} (space-separated): ").split()
            graph[node] = neighbors

        start_node = input("Enter the starting node for DFS: ")

        if start_node not in graph:
            print("Starting node is not in the graph!")
            return

        print("\nDepth-First Search starting from node", start_node)
        dfs(graph, start_node)

    except ValueError:
        print("Invalid input. Please enter valid numbers and nodes.")

if __name__ == "__main__":
    main()
```

OUTPUT:



```
Enter the number of nodes in the graph: 4
Enter the adjacency list:
Enter the node: A
Enter the neighbors of A (space-separated): B C
Enter the node: B
Enter the neighbors of B (space-separated): D
Enter the node: C
Enter the neighbors of C (space-separated): D
Enter the node: D
Enter the neighbors of D (space-separated):
Enter the starting node for DFS: A

Depth-First Search starting from node A
A B D C
```

RESULT: Thus, the Depth First Search program has beenimplementedsuccessfully.