

**EX.NO:2**

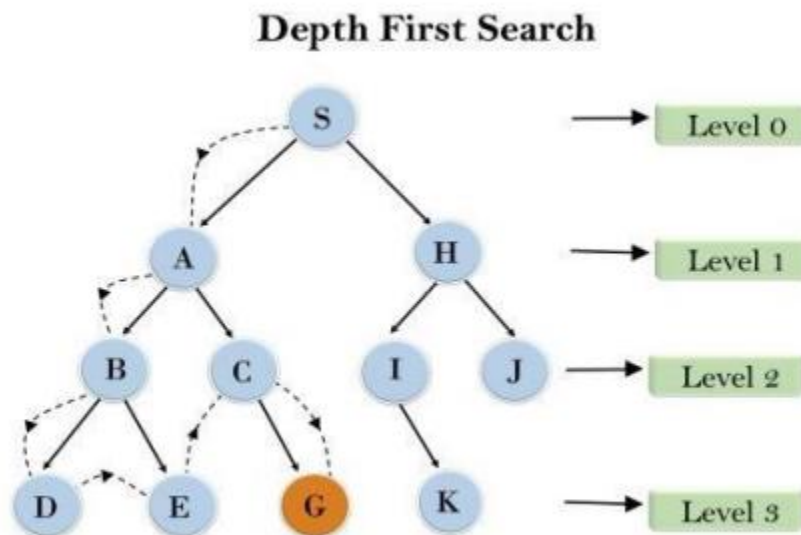
**DATE:4/9/2024**

**Reg.no:220701902**

## **DEPTH-FIRST SEARCH**

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

- Depth-first search (DFS) algorithm or searching technique starts with 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 by visiting different node of the tree.
- The algorithm, then backtracks or returns back from the dead end or last node towards the most recent node that is yet to be completely unexplored.
- The data structure (DS) which is being used in DFS Depth-first search 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 node are called block edges



## CODE:

```
def dfs_recursive(graph, start, visited=None):

    if visited is None:
        visited = set()
    visited.add(start)
    print(start)

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

graph = {
    'A': ['B', 'C'],
    'B': ['A', 'D', 'E'],
    'C': ['A', 'F'],
    'D': ['B'],
    'E': ['B', 'F'],
    'F': ['C', 'E']
}

print("DFS Recursive:")
dfs_recursive(graph, 'A')

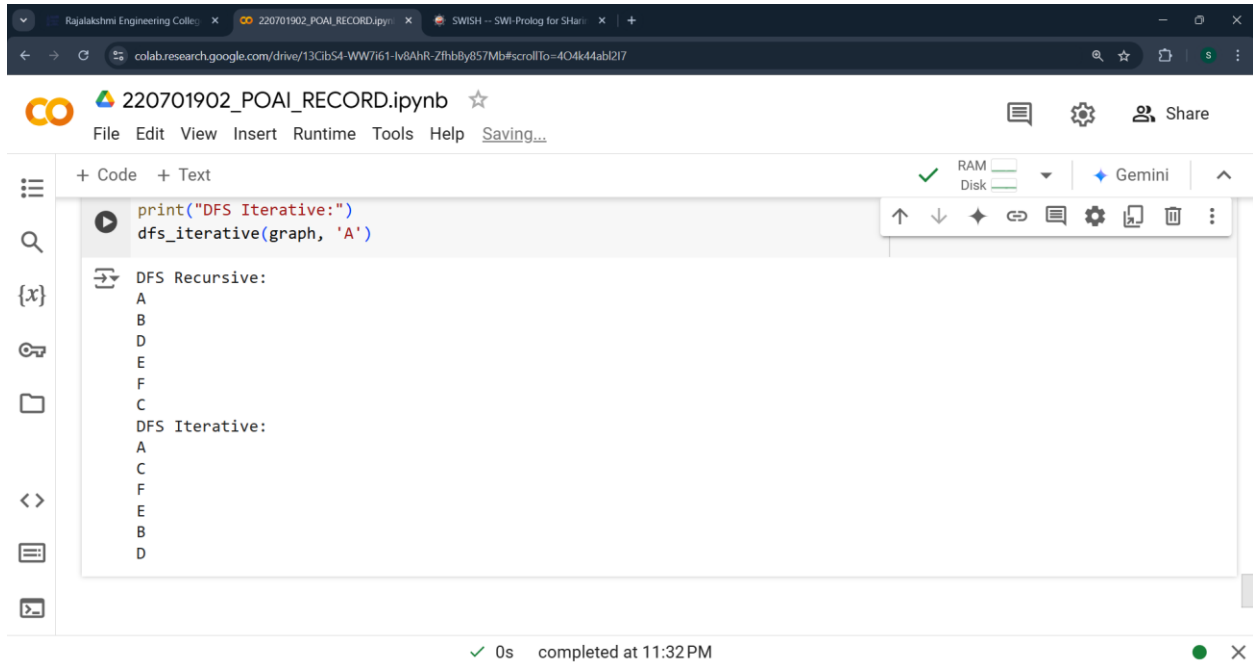
def dfs_iterative(graph, start):
    visited = set()
    stack = [start]

    while stack:
        vertex = stack.pop()
        if vertex not in visited:
            print(vertex)
            visited.add(vertex)
            stack.extend(neighbor for neighbor in graph[vertex] if
neighbor not in visited)

graph = {
    'A': ['B', 'C'],
    'B': ['A', 'D', 'E'],
    'C': ['A', 'F'],
    'D': ['B'],
    'E': ['B', 'F'],
    'F': ['C', 'E']
}

print("DFS Iterative:")
dfs_iterative(graph, 'A')
```

# OUTPUT:



The screenshot shows a Google Colab notebook interface. The browser tabs at the top include 'Rajalakshmi Engineering Colle...', '220701902\_POAI\_RECORD.ipynb', and 'SWISH -- SWI-Prolog for SHar...'. The address bar shows a Google Drive link. The notebook title is '220701902\_POAI\_RECORD.ipynb'. The menu bar includes 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', 'Help', and 'Saving...'. The left sidebar contains icons for file management and search. The main code area has a code cell with the following Python code:

```
print("DFS Iterative:")
dfs_iterative(graph, 'A')
```

Below the code cell, the output is displayed in a scrollable area. It shows the results of a Depth-First Search (DFS) algorithm. The output is as follows:

```
DFS Recursive:
A
B
D
E
F
C
DFS Iterative:
A
C
F
E
B
D
```

At the bottom of the notebook, a status bar indicates that the execution was successful: '✓ 0s completed at 11:32 PM'.