# 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.
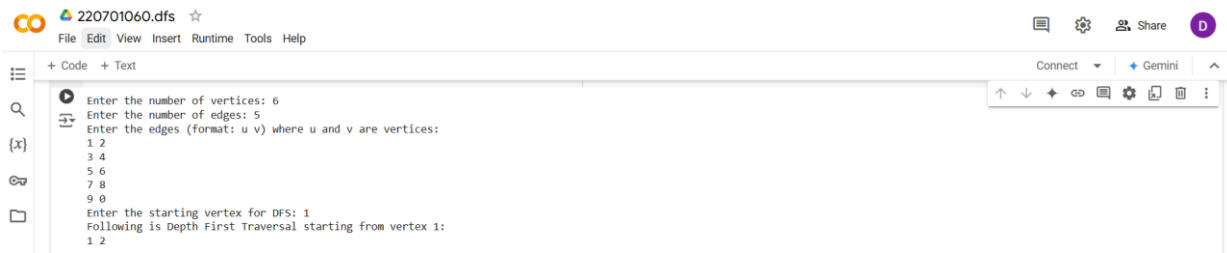

Depth First Search
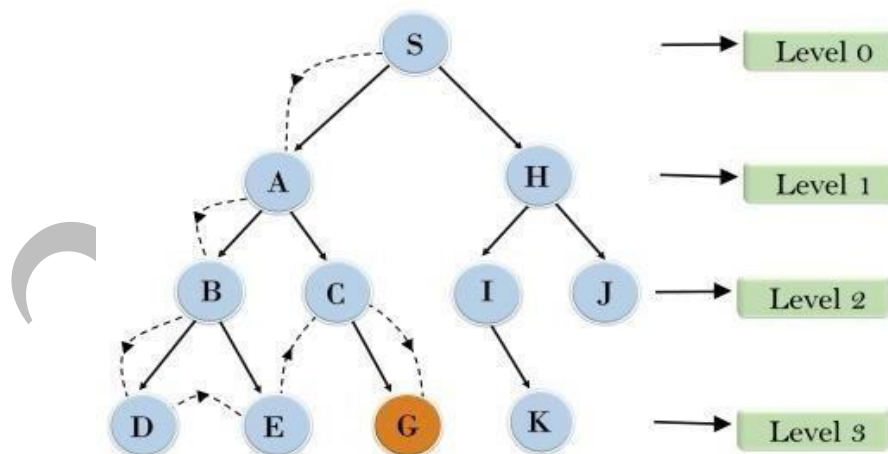
CODE:

```python
from collections import defaultdict
class Graph:
    def __init__(self):
        self.graph = defaultdict(list)
    def addEdge(self, u, v):
        self.graph[u].append(v)
    def DFSUtil(self, v, visited):
        visited.add(v)
        print(v, end=' ')
        for neighbour in self.graph[v]:
            if neighbour not in visited:
                self.DFSUtil(neighbour, visited)
    def DFS(self, v):
        visited = set()
        self.DFSUtil(v, visited)
if __name__ == "__main__":
    g = Graph()
    num_vertices = int(input("Enter the number of vertices: "))
    num_edges = int(input("Enter the number of edges: "))
    print("Enter the edges (format: u v) where u and v are vertices:")
    for _ in range(num_edges):
        u, v = map(int, input().split())
        g.addEdge(u, v)

    start_vertex = int(input("Enter the starting vertex for DFS: "))

    print(f"Following is Depth First Traversal starting from vertex {start_vertex}:")
    g.DFS(start_vertex)
```

OUTPUT:

```
CO    220701060.dfs  ☆                                              ⬛  ⚙  ⅏ Share  D
      File  Edit  View  Insert  Runtime  Tools  Help

☰     + Code  + Text                                         Connect  ▼   ✦ Gemini   ⌃

Q     ⬤   Enter the number of vertices: 6                    ↑  ↓  ✦  ⊝  ⬛  ⚙  ⬚  🗑  ⋮
      ⇥   Enter the number of edges: 5
          Enter the edges (format: u v) where u and v are vertices:
{x}         1 2
            3 4
            5 6
⊙↗          7 8
            9 0
☐         Enter the starting vertex for DFS: 1
          Following is Depth First Traversal starting from vertex 1:
            1 2
```

RESULT:

Thus, the Depth First Search program has been implemented successfully.

```
☰
{x}       ⬤   Enter the number of edges: 5
          ⇥   Enter the edges (format: u v) where u and v are vertices:
                                                            ↑  ↓  ✦  ⊝  ⬛  ⚙  ⬚  🗑  ⋮
```