DEPTH FIRST SEARCH

SOURCE CODE:

```
class Graph:
    def __init__(self, vertices):
        self.vertices = vertices
        self.graph = {i: [] for i in range(vertices)}
    def add_edge(self, u, v):
        self.graph[u].append(v)
        self.graph[v].append(u)

    def dfs(self, v, visited):
        print(v, end=" ")  # Print the current node
        visited[v] = True  # Mark the node as visited

        for neighbor in self.graph[v]:
            if not visited[neighbor]:
                self.dfs(neighbor, visited)
def depth_first_search():

    g = Graph(6)

    g.add_edge(0, 1)
    g.add_edge(0, 2)
    g.add_edge(1, 3)
    g.add_edge(1, 4)
    g.add_edge(2, 5)

    visited = [False] * g.vertices
```

```
    print("Depth First Search starting from vertex 0:")
    g.dfs(0, visited)
depth_first_search()
```

OUTPUT:

Depth First Search starting from vertex 0:

0 1 3 4 2 5