Implement detection of connected component using DFS

Connected components in a graph are the subgraphs in which all the vertices are connected to each other by at least one path.

Real-world examples and scenarios:

- **Social Networks:** In a social network, connected components can help identify groups of friends or communities that are closely connected to each other.
- **Computer networks:** In a computer network, connected components represent groups of devices that are directly or indirectly connected to each other. This information is crucial for designing efficient routing algorithms and detecting network vulnerabilities.
- **Transportation networks:** In a transportation network, connected components represent the clusters of cities that are reachable from each other. This knowledge is essential for planning efficient transportation systems and identifying potential bottlenecks.

Here are the steps to find the connected components using DFS:

1. Initialize a set or a list to store the connected components.
2. Iterate through all the vertices in the graph.
3. For each vertex, if it is not visited, perform a DFS from the vertex and mark all the vertices reachable from the vertex as visited.
4. Add the visited vertices to the connected component.
5. Repeat steps 3 and 4 until all the vertices are visited.

Python Code:

```
from collections import defaultdict
class Graph:
    def __init__(self, vertices):
        self.V = vertices
        self.graph = defaultdict(list)

    def add_edge(self, u, v):
        self.graph[u].append(v)
        self.graph[v].append(u)
    def DFS(self, v, visited):
        visited[v] = True
        cc = [v]
        for i in self.graph[v]:
            if not visited[i]:
                cc += self.DFS(i, visited)
        return cc

    def connected_components(self):
        visited = [False] * self.V
        cc = []
        for v in range(self.V):
            if not visited[v]:
                cc.append(self.DFS(v, visited))
        return cc

# Create a sample graph representing the social network
g = Graph(5)
g.add_edge(0, 1)
g.add_edge(1, 2)
g.add_edge(3, 4)

connected_components = g.connected_components()
print("Connected Components:", connected_components)
```