DFS/BFS

- DFS & BFS are search algorithms
- Run at a time complexity of $O(V+E)$.

Applications

1. finding bridges
2. finding longest path
3. finding shortest path in a undirected graph.

pseudo code for DFS

Mark visited so that child doesn't explore parent

```
fn dfs(node):
    if visited[node]: return      ← Avoid recomputation
    visited[node] = true          ← Mark visited

    for next in neighbors[node]:   ← Explore neighbors
        dfs(next)
```
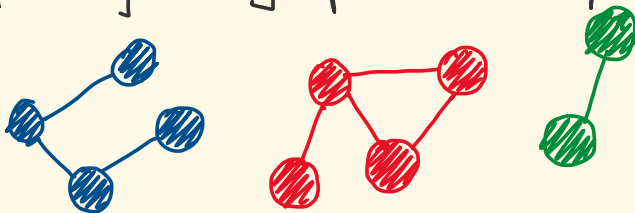
★ DFS is also called as flood fill

Connected components

- Splitting the graph into multiple components.

- Count no. of times u call the DFS/BFS algorithm for exploring the nodes.

Differences in BFS.

- DFS explores a particular branch till the end.
- BFS explores graph in layers.
- DFS uses a stack Data structure.
- BFS uses a queue Data structure.

pseudo code

```
fn BFS(node):

    q = [node]

    while q.size():
        cur = q.pop()

        for next in neighbors:
            if !visited[next]:
                q.push(next)
                visited[next] = T      ← mark visited before popping
    return
```