Experiment 6 :-

1) BFS (Breadth First Search)

```
BFS ( G, s)
  for each vertex v ∈ V[G] do
    explored [v] ← false
     d[v] ← ∞
  end for
    explored [s] ← true
    d[s] ← 0
  Q: queue data structure, initialized with s
  while Q ≠ ϕ do
      u ← remove vertex from front of Q
    for each v adjacent to u do
       if not explored[v] then
          explored [v] ← true
           d[v] ← d[u] + 1
     insert v to the end of Q
  end if
  end for
   end while
```

* where G is Graph and s is source vertex

classmate
Date
Page

2) DFS (

→ Iterative

```
DFS(G, S)
    let S be stack
    S.push(s)
    mark s as visited
    while (S is not empty)
        pop a vertex from stack to visit next
            v = S.top()
            S.pop()
    for all neighbours w of v in Graph G:
        if w is not visited:
                S.push(w) mark w as visited
    end of if, while, DFS()
```

→ Recursive

```
DFS (G, S)
    mark s as visited
    for all neighbours w of s in Graph G:
        if w is not visited
            DFS (G, w)
```