# Search Algorithms

GraphSearch($V, E, s, searchtype$)

```
    // Input: Vertex set V, edge set E, starting vertex s, searchtype as "BFS" or "DFS"
    // Output: Path from s to every vertex in V
    // Running Time: O(|V| + |E|)
 1  for v in V
 2      color[v] ← White
 3      d[v] ← ∞
 4      π[v] ← [] // π[v] is parent of v
 5  color[s] ← Gray; d[s] ← 0; π[s] ← []
 6  if searchtype = "BFS"
 7      Q = queue()
 8  else
 9      Q = stack()
10  Q.push(s)
11  while not Q.empty()
12      u ← Q.pop()
13      for each v in Adj[u]
14          if color[v] = White // v not in Q
15              color[v] ← Gray
16              d[v] ← d[u] + 1; π[v] ← u
17              Q.push(v)
18      color[u] ← Black
```

The recursive version of DFS computes discovery times $d$ and finishing times $f$ for each vertex:

DFS($V, E$)

```
    // Input: Vertex set V, edge set E
    // Output: Discovery and finish time for each vertex in V
    // Running Time: O(|V| + |E|)
1   for v in V
2       color[v] ← White
3       π[v] ← [] // π[v]isparentofv
4   time ← 0
5   for u ∈ V
6       if color[u] = White
7           DFS-Visit(u)
```

DFS-VISIT($u$)

```
 1  color[u] ← Gray
 2  time ← time + 1
 3  d[u] ← time
 4  for v in Adj(u)
 5        if color[v] = White
 6              π[v] ← u
 7              DFS-VISIT(v)
 8  color[u] ← Black
 9  time ← time + 1
10  f[u] ← time
```