Graph Traversal

A graph traversal algorithm inspects all vertices and all edges of the graph

Depth First Search (DFS)

- searches deeper in the graph whenever possible

Breadth First Search (BFS)

- explores the edges of a vertex *s* to discover every vertex that is reachable from *s*

Assume a graph has *n* vertices and *e* edges

DFS Algorithm

```
DFS Driver ()
   for each vertex V in the graph
      if V is not visited
         DFS(V)
      endif
   endfor
```

The driver is needed in case the graph is unconnected

```
DFS (V)
   visit V and mark V as visited
   for each neighbor W of V
      if W is not visited
         DFS(W)
      endif
   endfor
```

DFS Running Time

```
DFS Driver ()
    for each vertex V in the graph
        if V is not visited
            DFS(V)
        endif
    endfor

DFS (V)
    visit V and mark V as visited
    for each neighbor W of V
        if W is not visited
            DFS(W)
        endif
    endfor
```

Executed once for every vertex: n times

Executed once for every vertex: n times

Executed for every edge: e times

Undirected:
$n + n + 2e = 2n + 2e = O(n+e)$

Directed:
$n + n + e = 2n + 2e = O(n+e)$

BFS Algorithm

```
BFS Driver ()
    for each vertex V in the graph
        if V is not visited
            DFS(V)
        endif
    endfor
```

The driver is needed in case the graph is unconnected

```
BFS (V)
    visit V and mark V as visited
    enqueue V
    while the queue is not empty
        W = dequeue
        for each neighbor P of W
            if P is not visited
                visit P and mark P as visited
                enqueue P
            endif
        endfor
    endwhile
```

BFS Running Time

```
BFS Driver ()
    for each vertex V in the graph
        if V is not visited
            DFS(V)
        endif
    endfor

BFS (V)
    visit V and mark V as visited
    enqueue V
    while the queue is not empty
        W = dequeue
        for each neighbor P of W
            if P is not visited
                visit P and mark P as visited
                enqueue P
            endif
        endfor
    endwhile
```

Executed once for every vertex: n times

A vertex is marked as visited only once: n times

Executed for every edge: e times

Undirected:

$n + n + 2e = 2n + 2e = O(n+e)$

Directed:

$n + n + e = 2n + 2e = O(n+e)$