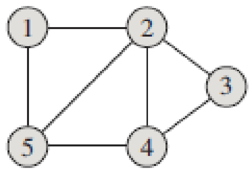
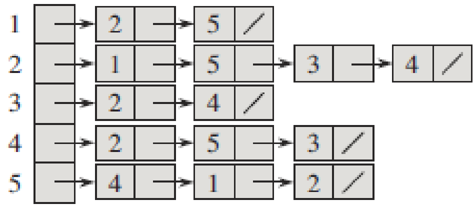
Graph Algorithms

Adjacency List

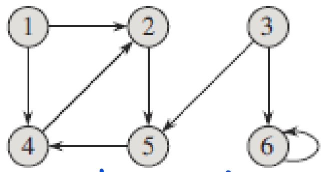
Adjacency matrix



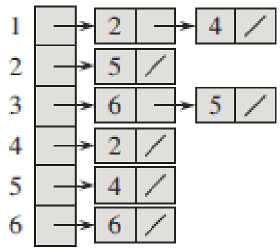
Undirected graph



	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

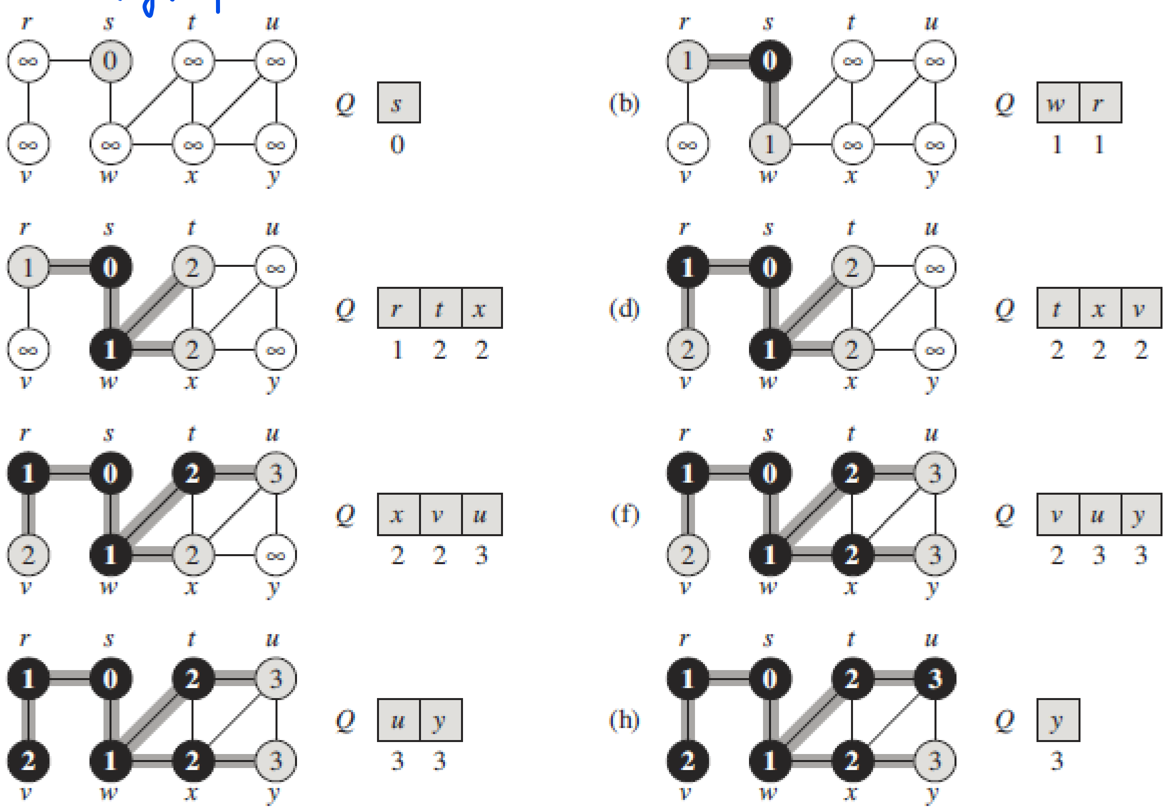


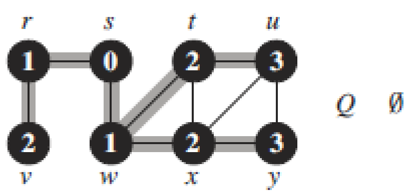
Directed graph

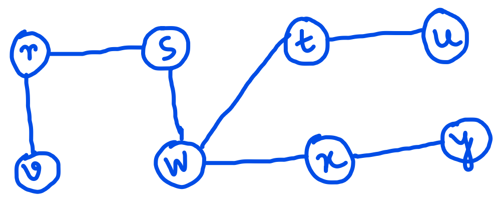


	1	2	3	4	5	6
1	0	1	0	1	0	0
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	1	0	0	0	0
5	0	0	0	1	0	0
6	0	0	0	0	0	1

BFS: (Breadth First Search)







BFS tree

BFS(G, s)

```
O(V)    for each vertex u ∈ G.V − {s}
            u.color = WHITE
            u.d = ∞
            u.π = NIL
Const.  s.color = GRAY
        s.d = 0
        s.π = NIL
        Q = ∅
        ENQUEUE(Q, s)
        while Q ≠ ∅
            u = DEQUEUE(Q)
            for each v ∈ G.Adj[u]
                if v.color == WHITE
                    v.color = GRAY
                    v.d = u.d + 1
                    v.π = u
                    ENQUEUE(Q, v)
            u.color = BLACK
```

ENQUEUE & DEQUEUE → $O(1)$

E → No. of edges present in G.

Total time spent in scanning adjacency list $= O(E)$

$\therefore \underline{\underline{O(V+E)}}$