Graph Algorithms

A graph is a set of connected objects (vertices **V**)

Connections between vertices are called edges and represented by the notation **E**

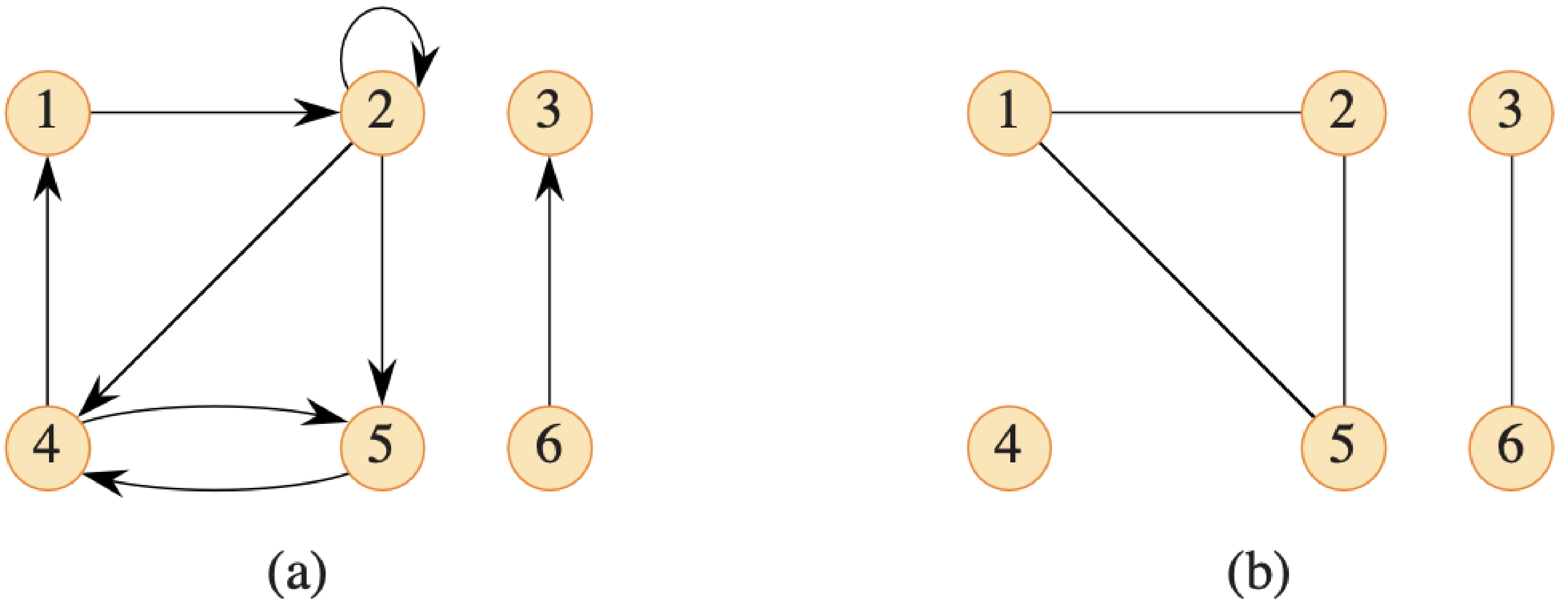


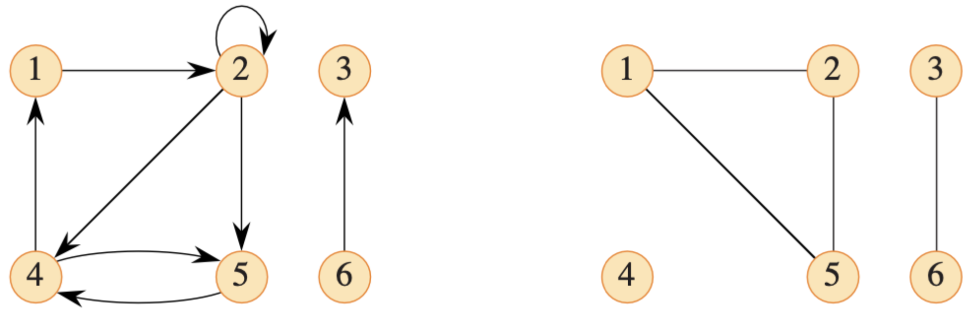
Figure 1: Directed and undirected graphs

Graphs can be directed as in (a) and undirected as in (b)

Graph Applications

The world is a network of connected objects

- Communication networks
- Links between websites
- Social Networks
- Gene ontologies in bioinformatics
- Supply chain networks in e-commerce and logistics
- Brain neural networks



(a) (b)

A directed graph $G = (V, E)$

E is a set of binary relations between every ordered pair in V

For instance, (1,2) and (2,1) are two different pairs in E

The **out-degree** of a vertex is the number of edges leaving it

The **in-degree** of a vertex is the number of edges entering it

The **degree** of a vertex is the out-degree + the in-degree

In undirected graphs, pairs in E are not ordered

We can define an edge as a set $\{u, v\}$, where $u, v \in V$ and $u \neq v$

The **degree** of a vertex in an undirected graph is the number of edges on it

A path of length k from a vertex u to u' is a sequence $< v_0, v_1, ..., v_k >$ such that $u = v_0$ and $u' = v_k$

In a directed graph, a path $< v_0, v_1, ..., v_k >$ forms a **cycle** if $v_0 = v_k$

An *undirected* graph is **connected** if every vertex is reachable from all other vertices

A connected component is a connected subgraph

Graph (b), from Figure 1, has three connected components, including 4

A graph is connected if it has exactly one connected component

A *directed* graph is **strongly connected** if every two vertices are reachable from each other

Graph (a) has the following strongly connected components $\{1, 2, 4, 5\}$, $\{3\}$, and $\{6\}$

Graph Representation

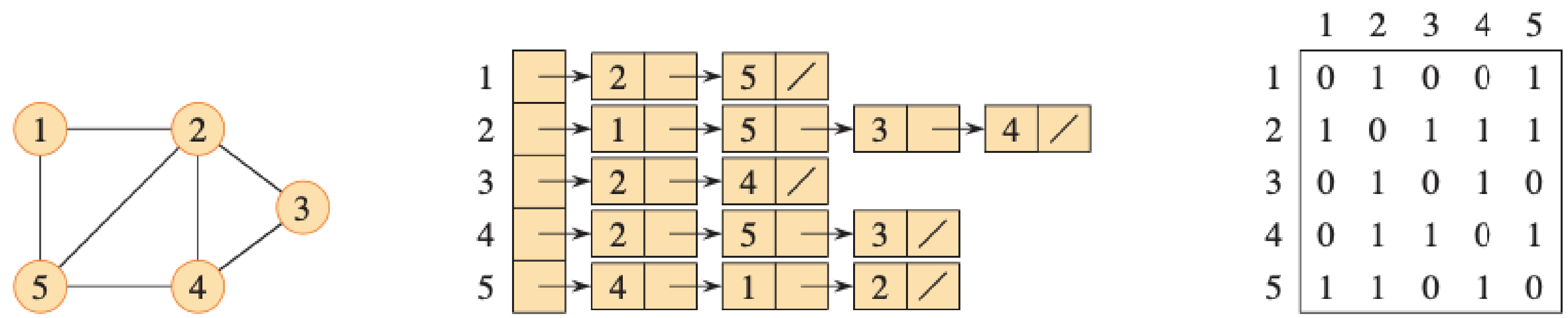
The adjacency list provides compact representation of a graph

It consists of an array Adj of $|V|$ lists, one for each vertex

The adjacency list $Adj[u]$ contains all the vertices v such that there is an edge $(u, v) \in E$

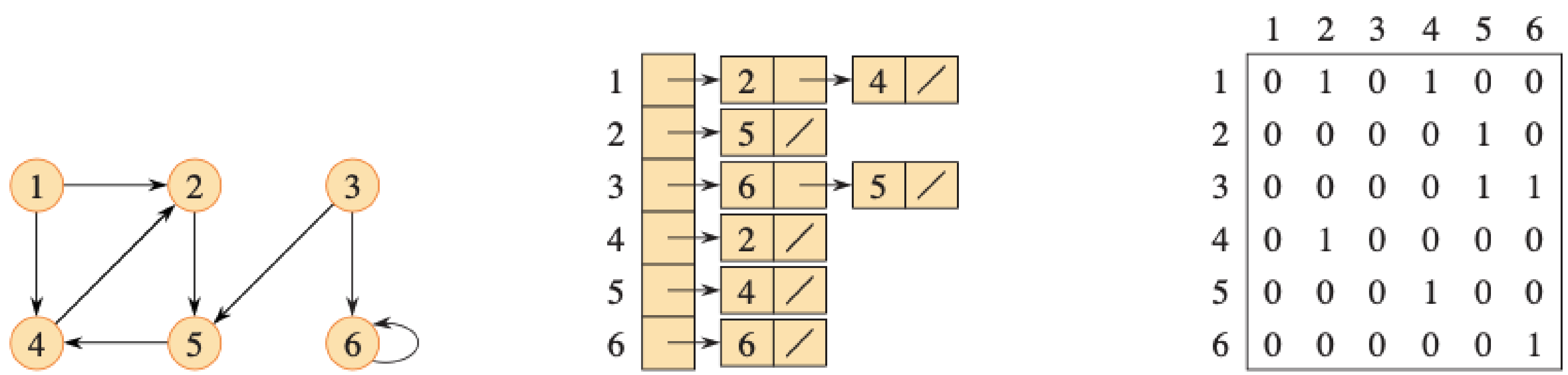
In a directed graph, the sum of the lengths of all the adjacency lists is $|E|$

In an undirected graph, the sum of the lengths of all the adjacency lists is $2|E|$, since each edge (u, v) in an adjacency list exists in another list in the form of (v, u)



(a) (b) (c)

Figure 20.1 Two representations of an undirected graph. **(a)** An undirected graph G with 5 vertices and 7 edges. **(b)** An adjacency-list representation of G. **(c)** The adjacency-matrix representation of G.



(a) (b) (c)

Figure 2: Two graph representations

The adjacency list size is $(V + E)$

Adjacency lists can also represent weights in edges by a weight function $w : E \rightarrow \mathbb{R}$

For example, $w(u, v) = 0.5$

A vertex or an edge may maintain attributes

A vertex v with the attribute d can be written as $v.d$

An edge (u, v) with the attribute d can be written as $(u, v).d$

The adjacency matrix is another representation for graphs

The matrix size is $|V| \times |V|$

Each item a_{ij} has a binary value:

$$a_{ij} = \begin{cases} 1 & \textbf{if } (i, j) \in E \ , \\ 0 & \textbf{otherwise} \ . \end{cases}$$

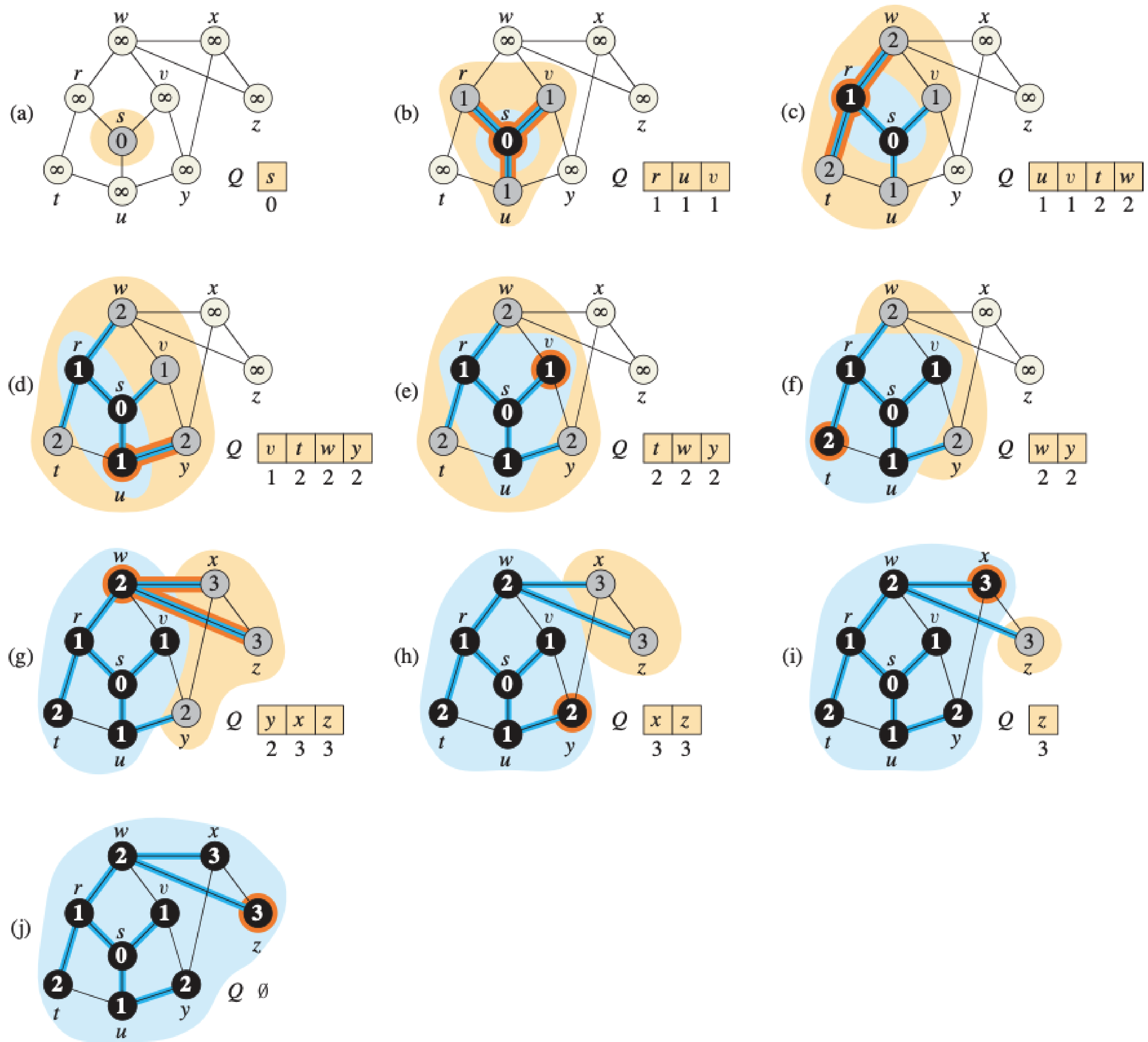
Observe the symmetry along the main diagonal of the adjacency matrix, which suggests reduced memory need

Breadth-first search

Given a source vertex s, the algorithm discovers every vertex reachable from s

The algorithm works on both directed and undirected graphs

```
BFS(G, s)
for each vertex u ∈ G.V − {s}
    u.color = WHITE
    u.d = ∞
    u.π = NIL
s.color = GRAY
s.d = 0
s.π = NIL
Q = ∅
ENQUEUE(Q, s)
while Q ≠ ∅
    u = DEQUEUE(Q)
    for each vertex v in G.Adj[u]    // search the neighbors of u
        if v.color == WHITE          // is v being discovered now?
            v.color = GRAY
            v.d = u.d + 1
            v.π = u
            ENQUEUE(Q, v)            // v is now on the frontier
    u.color = BLACK                  // u is now behind the frontier
```


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)
Q s 0
Q r u v 1 1 1
Q u v t w 1 1 2 2
Q v t w y 1 2 2 2
Q t w y 2 2 2
Q w y 2 2
Q y x z 2 3 3
Q x z 3 3
Q z 3
Q Ø