2.2 Graph Representation Methods

Graph representation methods describe how a graph is stored in computer memory so that graph algorithms (BFS, DFS, Dijkstra, etc.) can be executed efficiently.

The two main graph representation methods are:

1. Adjacency Matrix
2. Adjacency List
3. Edge List
4. Incidence Matrix

1. Adjacency Matrix

Definition

An adjacency matrix is a 2D array (matrix) of size $n \times n$, where $n$is the number of vertices in the graph.

$$A[i][j] = \begin{cases} 1 & \text{if there is an edge from } i \text{ to } j \\ 0 & \text{otherwise} \end{cases}$$

Example

Consider an undirected graph:

A — B

| |

C — D

Vertices: A, B, C, D

Adjacency Matrix:

	A	**B**	**C**	**D**
A	0	1	1	0
B	1	0	0	1
C	1	0	0	1
D	0	1	1	0

Directed Graph Case

- $A[i][j] = 1 \rightarrow$ edge from $i$to j
- $A[j][i]$may be 0

Weighted Graph Case

- Matrix stores **weights instead of 1**

Example:

$$A[i][j] = \text{weight of edge}$$

Time and Space Complexity

- **Space:** $O(n^2)$
- **Edge lookup:** $O(1)$

Advantages

✓ Very simple to implement
✓ Fast edge existence checking
✓ Useful for dense graphs

Disadvantages

✗ High memory usage
✗ Inefficient for sparse graphs

Applications

- Dense networks
- Small graphs
- Graph algorithms in theory

2. Adjacency List

Definition

An adjacency list stores a list of neighbors for each vertex.

Example

Graph:

A — B

| |

C — D

Adjacency List:

A → B, C

B → A, D

C → A, D

D → B, C

Directed Graph Case

- Only outgoing neighbors stored

Weighted Graph Case

- Store (neighbor, weight) pairs

Example:

A → (B,5), (C,3)

Time and Space Complexity

- **Space:** $O(V + E)$
- **Edge lookup:** $O(\text{degree})$

Advantages

✓ Memory efficient
✓ Ideal for sparse graphs
✓ Used in BFS, DFS, Dijkstra

Disadvantages

✕ Slower edge lookup
✕ Slightly complex implementation

Applications

- Social networks
- Web graphs
- Large-scale graphs

3. Edge List

Definition

An edge list stores all edges as pairs (or triples for weighted graphs).

Example

Edge List:

(A, B)

(A, C)

(B, D)

(C, D)

Weighted:

(A, B, 5)

Complexity

- **Space:** $O(E)$
- **Edge lookup:** $O(E)$

Advantages

✓ Very simple
✓ Easy to read and store

Disadvantages

✗ Very slow for traversal
✗ Not suitable for most algorithms

Applications

- Kruskal's algorithm
- Graph input/output

4. Incidence Matrix

Definition

An incidence matrix is a matrix where:

- Rows represent vertices
- Columns represent edges

Example

Graph with edges e1, e2:

	e1	e2
A	1	0
B	1	1
C	0	1

Directed Graph Case

- +1 for outgoing edge
- −1 for incoming edge

Complexity

- **Space:** $O(V \times E)$

Advantages

✓ Useful in mathematical analysis

Disadvantages

✕ Very high space usage
✕ Rarely used in practice

Applications

- Network flow theory
- Electrical circuits