Theoretical Question

What are the differences between Adjacency List and Adjacency Matrix representation of a Graph?
Graphs can be represented in memory using different methods.
The two most common representations are Adjacency List and Adjacency Matrix.
Each representation has its own advantages and disadvantages depending on the type of graph and its usage.

1. Adjacency List

Definition

An Adjacency List represents a graph as an array (or list) of lists.
Each vertex has a list that contains all the vertices directly connected to it by edges.

Characteristics

Uses less memory for graphs with fewer edges.
Efficient for iterating over neighbors of a vertex.
More suitable for sparse graphs (graphs with fewer edges).
Adding a new edge is relatively easy.

Time Complexity

Checking if an edge exists between two vertices: O(n) in the worst case.
Traversing all adjacent vertices: O(degree of vertex).

Advantages

Memory efficient.
Flexible and scalable.
Faster traversal of connected vertices.

Disadvantages

Slower edge lookup compared to adjacency matrix.
More complex implementation.

2. Adjacency Matrix

Definition

An Adjacency Matrix represents a graph using a 2D array.
If there is an edge between vertex i and vertex j, the value at matrix[i][j] is 1 (or the weight of the edge), otherwise it is 0.

Characteristics

Uses a fixed amount of memory regardless of the number of edges.
Simple and easy to understand.
Suitable for dense graphs (graphs with many edges).

Time Complexity

Checking if an edge exists between two vertices: O(1).
Traversing all adjacent vertices: O(n).

Advantages

Fast edge lookup.
Simple implementation.
Useful for dense graphs.

Disadvantages

Wastes memory for sparse graphs.
Not efficient for large graphs with few edges