

Question:

What are the differences between **Adjacency List** and **Adjacency Matrix** representations of a graph?

Adjacency List

An adjacency list represents a graph as an array (or list) of lists. Each index represents a vertex, and each list at that index contains the vertices adjacent to it.

Uses:

- Suitable for large graphs with a small number of edges (Sparse Graphs).
- Commonly used when memory efficiency is important.

Pros:

- Requires less memory compared to adjacency matrix.
- Efficient for iterating over neighbors of a vertex.
- Faster to add new edges.

Cons:

- Checking whether a specific edge exists takes more time ($O(V)$).
 - Implementation is more complex than adjacency matrix.
-

Adjacency Matrix

An adjacency matrix represents a graph using a 2D array where rows and columns represent vertices. A value of 1 (or true) indicates the presence of an edge, and 0 (or false) indicates no edge.

Uses:

- Suitable for small graphs or dense graphs.
- Used when fast edge lookup is required.

Pros:

- Easy to implement and understand.
- Checking whether an edge exists takes constant time ($O(1)$).

Cons:

- Consumes a large amount of memory, especially for large graphs.
- Inefficient for sparse graphs.