Graph Representation: Data Structures and Examples

Your Name

November 7, 2024

1 Introduction

Graphs are fundamental data structures used to represent relationships between objects. They consist of vertices (or nodes) and edges (connections between nodes). This article explores various data structures for graph representation, including adjacency matrices, adjacency lists, and edge lists.

2 1. Adjacency Matrix

An adjacency matrix is a 2D array used to represent a graph. The rows and columns represent the vertices, and the entries indicate whether pairs of vertices are adjacent.

2.1 Example

Consider a simple graph with 4 vertices: A, B, C, and D.

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

In this matrix:

- A is connected to B and C.
- B is connected to A and D.
- C is connected to A and D.
- D is connected to B and C.

2.2 Complexity

The space complexity of an adjacency matrix is $O(V^2)$, where V is the number of vertices. This representation is efficient for dense graphs.

3 2. Adjacency List

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

3.1 Example

Using the same graph as before, the adjacency list would look like this:

```
A: B, C
B: A, D
C: A, D
D: B, C
```

3.2 Complexity

The space complexity of an adjacency list is $O(V + E)$, where E is the number of edges. This representation is more efficient for sparse graphs.

4 3. Edge List

An edge list is a collection of edges, where each edge is represented as a pair of vertices.

4.1 Example

For the graph in question, the edge list would be:

```
(A, B)
(A, C)
(B, D)
(C, D)
```

4.2 Complexity

The space complexity of an edge list is $O(E)$. This representation is simple and useful for algorithms that process edges directly.

5 4. Comparison of Data Structures

Data Structure	Space Complexity	Best Use Case	Drawbacks
Adjacency Matrix	$O(V^2)$	Dense graphs	Wastes space for sparse graphs
Adjacency List	$O(V + E)$	Sparse graphs	Slower for checking edge existence
Edge List	$O(E)$	Edge-centric algorithms	Inefficient for adjacency queries

Table 1: Comparison of Graph Representation Data Structures