Linked Representation of a Graph

Md. Maskaowat Ahsan Hasby

Student ID: 2410876132

Dept. of Computer Science and Engineering

University of Rajshahi

Contents:

Introduction of Graphs

Definition

- **Graphs:**

A Graph **G** is a non-linear data structure defined by an ordered pair **(V, E)** where:

- **V (Vertices):** A set of nodes representing data entities.
- **E (Edges):** A set of connections between pairs of vertices.

Memory Representations

1. Sequential Representation

- Uses an **Adjacency Matrix** (2D Array).

2. Linked Representation

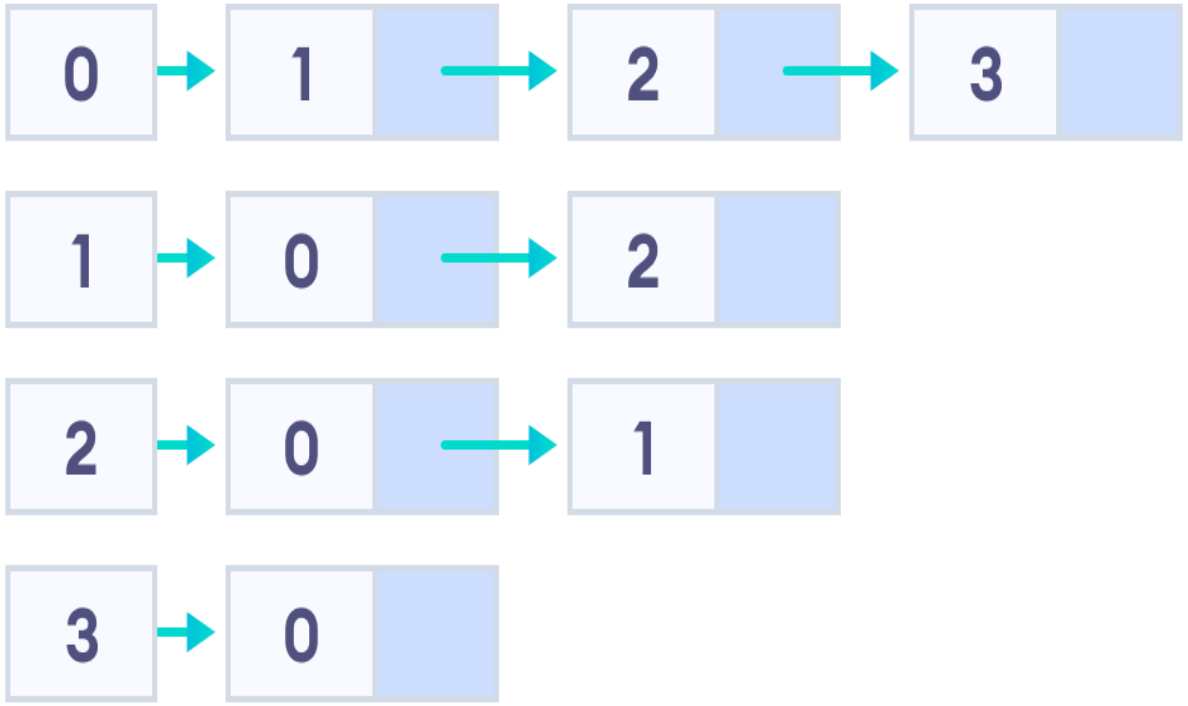
- Uses **Adjacency Lists** (Linked Lists).

Linked Representation Overview

The Concept

- In linked representation, each node in G is followed by its **adjacency list**.
- This list contains all its "successors" or "neighbors".
- This representation uses two specific lists:

1. Node list (NODE)
2. Edge list (EDGE)



Data Structure: The Node List, Edge List

- **Node List:** Each element in the **NODE** list corresponds to a vertex in the graph.

It is a record containing three fields:

1. NODE: Name or key value of the node.

2. NEXT: Pointer to the next node in the list.

3. ADJ: Pointer to the first element in the adjacency list (maintained in EDGE list).

Node record structure below:

* Note: Records may also store INDEG, OUTDEG, or STATUS.

- **Edge List:** Each element in the EDGE list corresponds to an edge of G.

It is a record containing two main fields:

1. DEST: Points to the location of the destination node in the NODE list.

2. LINK: Links together edges with the same initial node (the next neighbor).

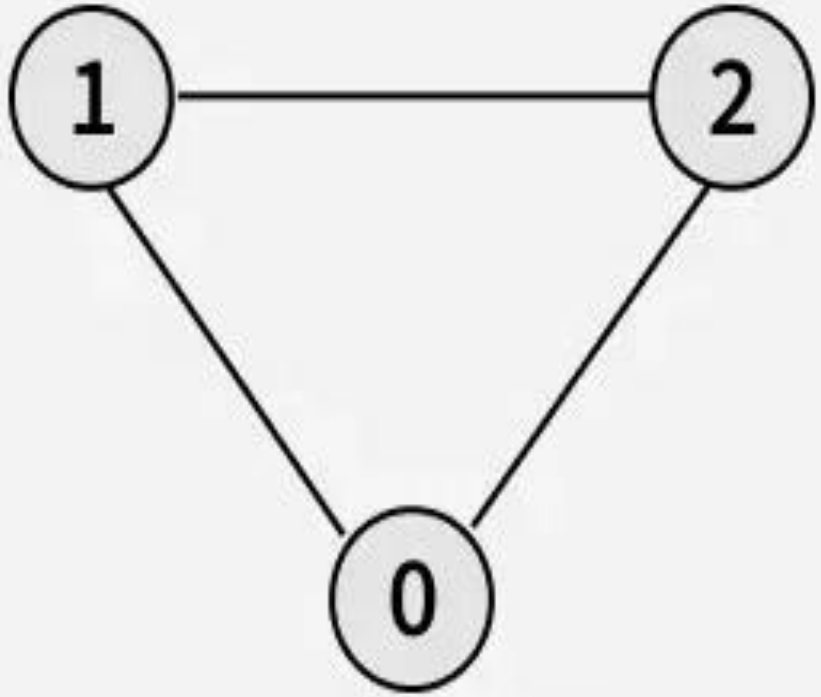
Edge record structure below:

* Records may also store edge weights or labels.

01 Step

Initialize adjacency list with V vertices

(Undirected Graph)



Undirected Graph

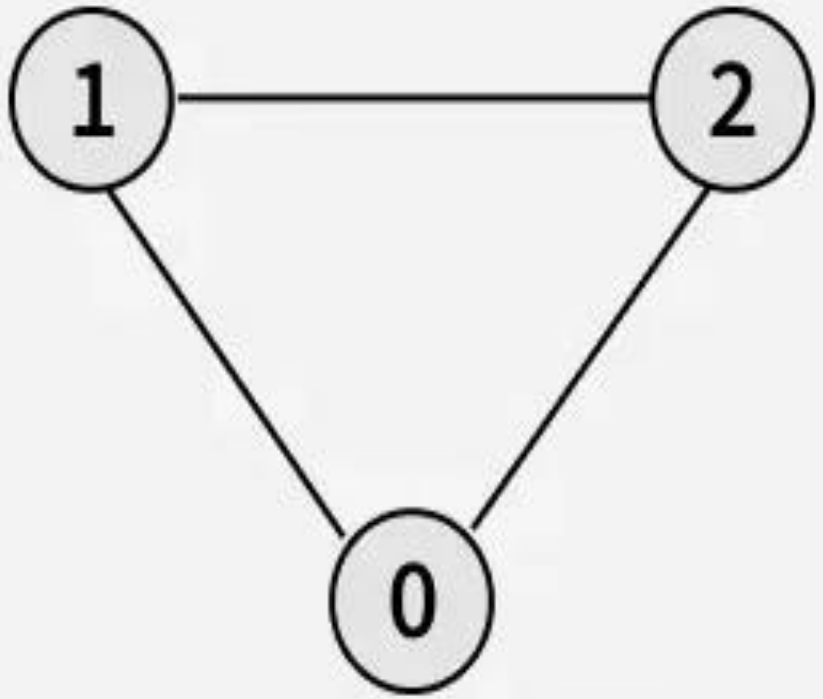
Adjacency List

u	v
1	0
1	2
2	0

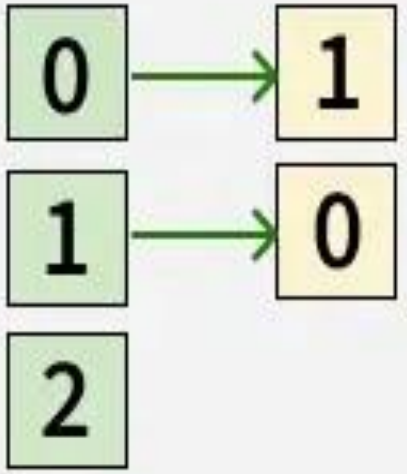
Edges

Graph Representation of Undirected graph to Adjacency List

02 Step | Adding (1, 0) and (0, 1) edge to adjacency list



Undirected Graph



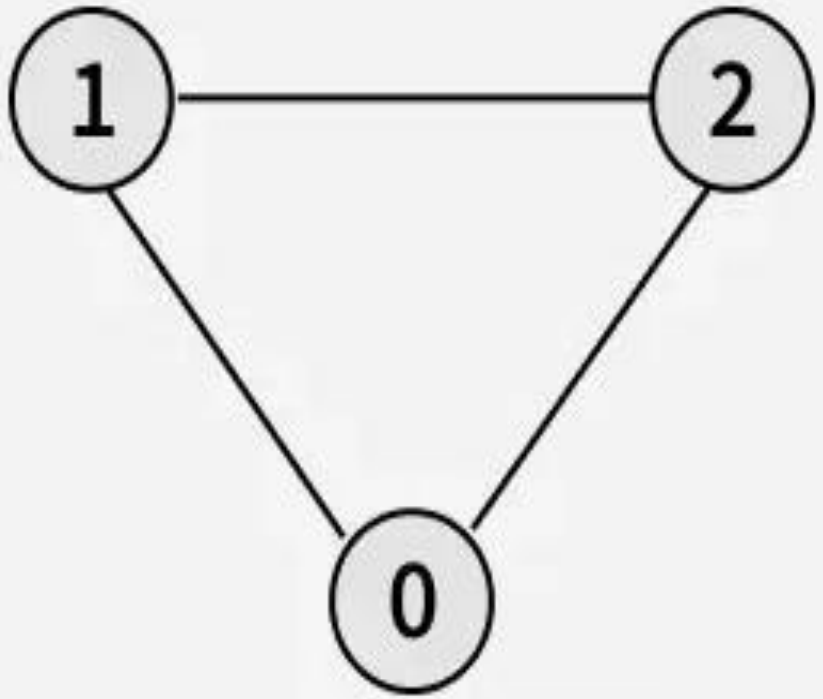
Adjacency List

u	v
1	0
1	2
2	0

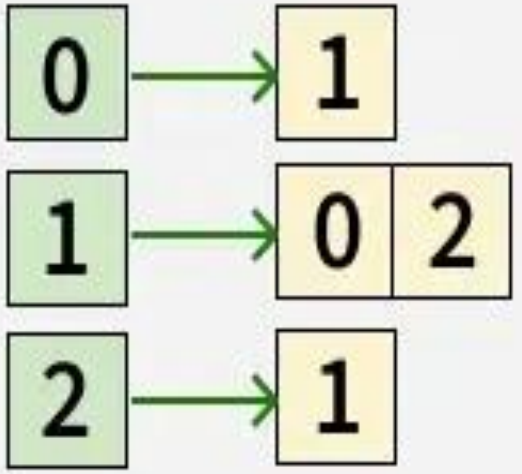
Edges

Graph Representation of Undirected graph to Adjacency List

03 Step | Adding (1, 2) and (2, 1) edge to adjacency list



Undirected Graph



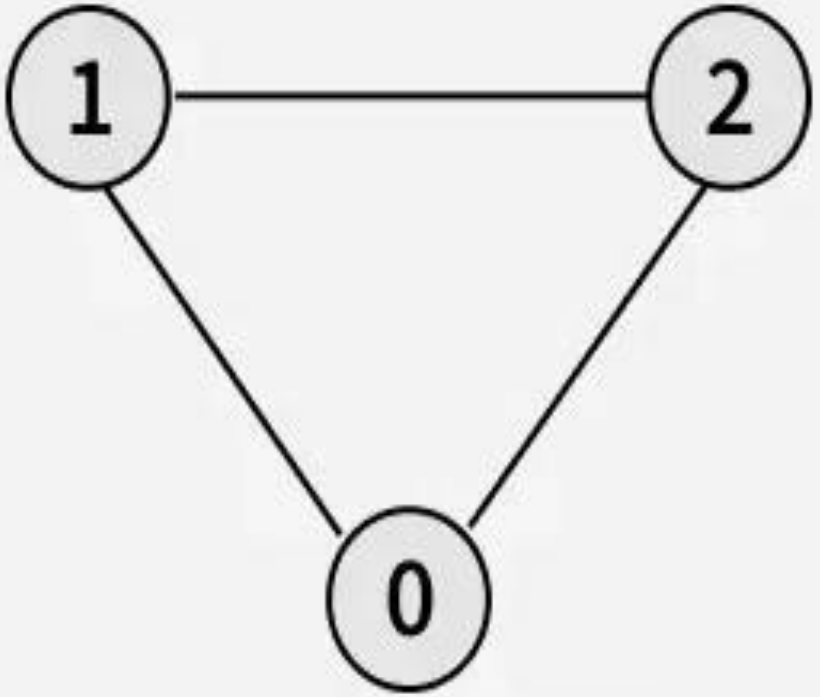
Adjacency List

u	v
1	0
1	2
2	0

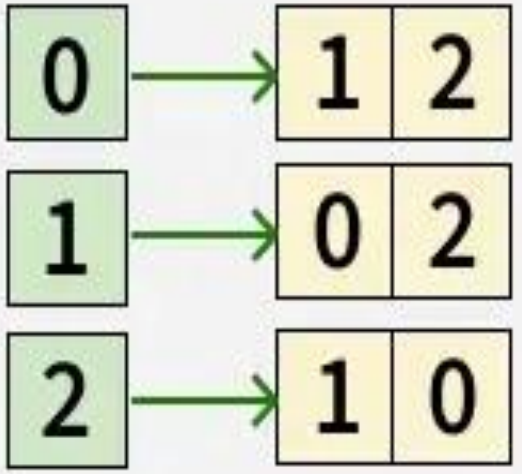
Edges

Graph Representation of Undirected graph to Adjacency List

04 Step | Adding (2, 0) and (2, 0) edge to adjacency list



Undirected Graph



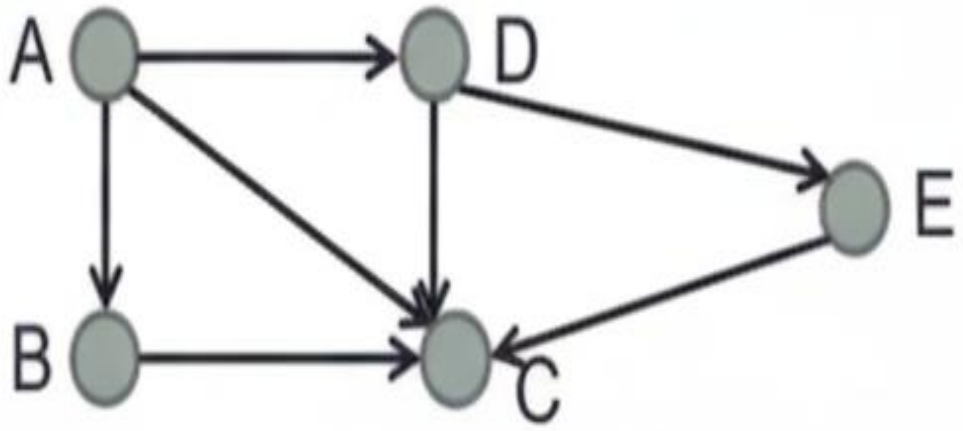
Adjacency List

u	v
1	0
1	2
2	0

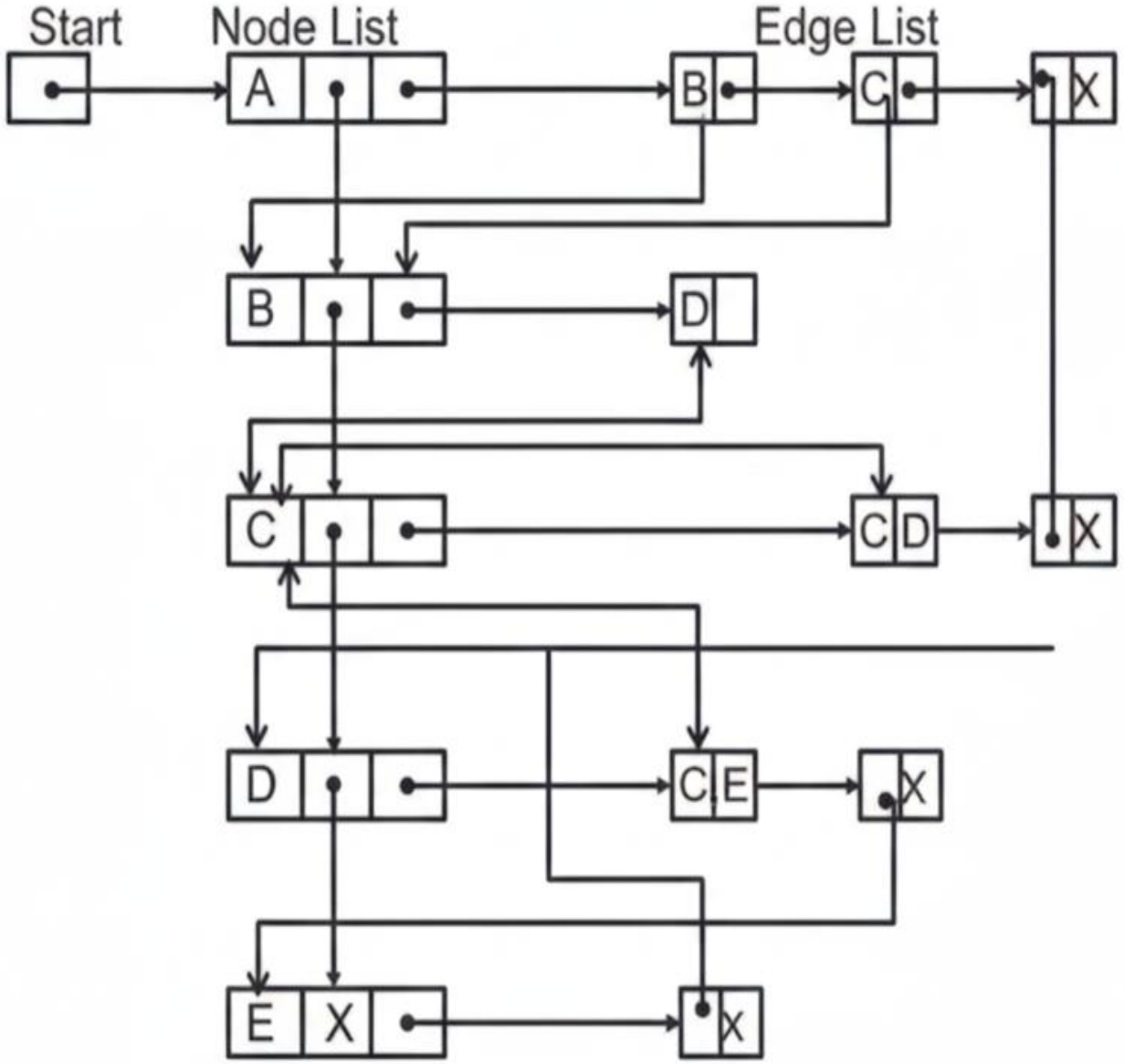
Edges

Graph Representation of Undirected graph to Adjacency List

For Directed Graph:



Node	Adjacency List
A	B, C, D
B	C
C	
D	X, E
E	C



Applications of the Adjacency List:

• **Graph algorithms:** Many graph algorithms like Dijkstra's algorithm, Breadth First Search, and Depth First Search perform faster for adjacency lists to represent graphs.

• The most commonly used representation of graph

• Allows easy traversal of all edges.

Thank You

For your patience