Graphs notes:

1.Simple graph: no self loops, only one edge between two nodes

2.If N is no of vertices, max no of edges= N(N-1); directed graph

3.If N is no of vertices, max no of edges= [N(N-1)]/2; undirected graph (assuming no multiedge or self loop)

4.Dense-> too many edges. Sparse-> few edges.

5.Normally, use matrix if dense and list if sparse.

6.Path- A sequence of vertices where each adjacent pair is connected by an edge.

7.Simple Path- A path in which no vertices (and thus no edges) are repeated.

8.Walk- A sequence of vertices where each adjacent pair is connected by an edge.

9.Path- A walk in which no vertices are repeated.

10.Trail- A walk in which no edges are repeated.

11. Strongly connected graphs- If there is a path from any vertex to any other vertex.

12.Closed walk: Starts and ends at same vertex

13.Simple cycle: No repetition other than start and end.

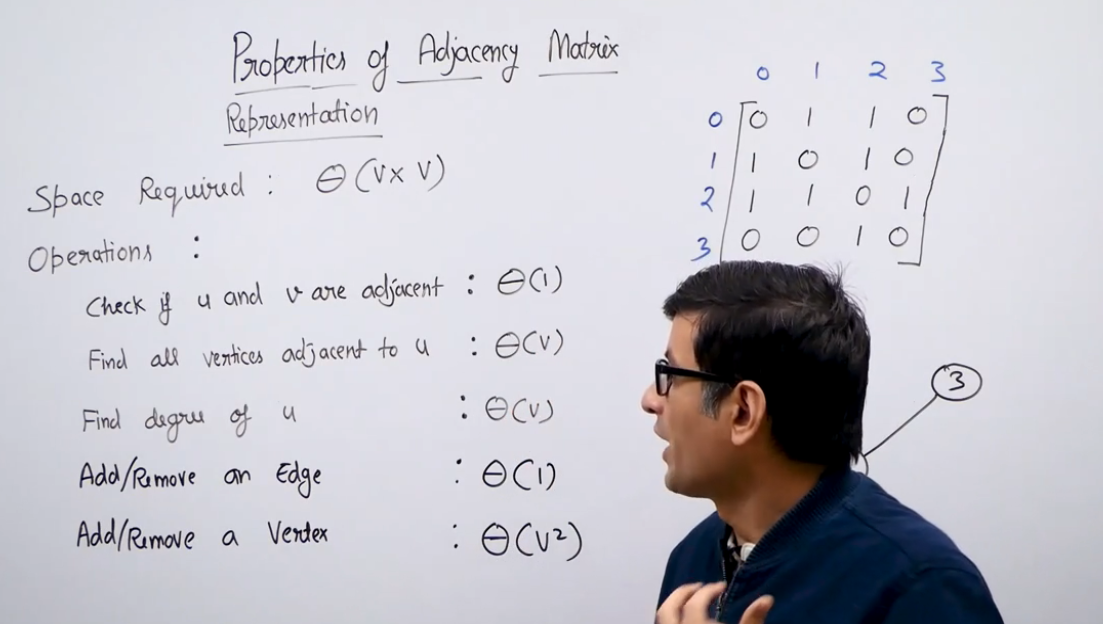
14.Acyclic Graph: A graph with no cycle

15.Adjacency Matrix:

if edge between vertex i to j => 1 matrix

else 0 in matrix

Undirected graph: Symmetric matrix, square matrix



Adjacency matrix causes redundancy and it not very preferred for sparse graphs.

**Adjacency List:**

Array of lists. Lists can be linked lists or dynamic arrays or any other data structure.

Need array to store index