

Dense Graph 🡪 This is something which has many edges

We should use adjacency matrix

Sparse Graph 🡪 Graph is sparse when there are less edges

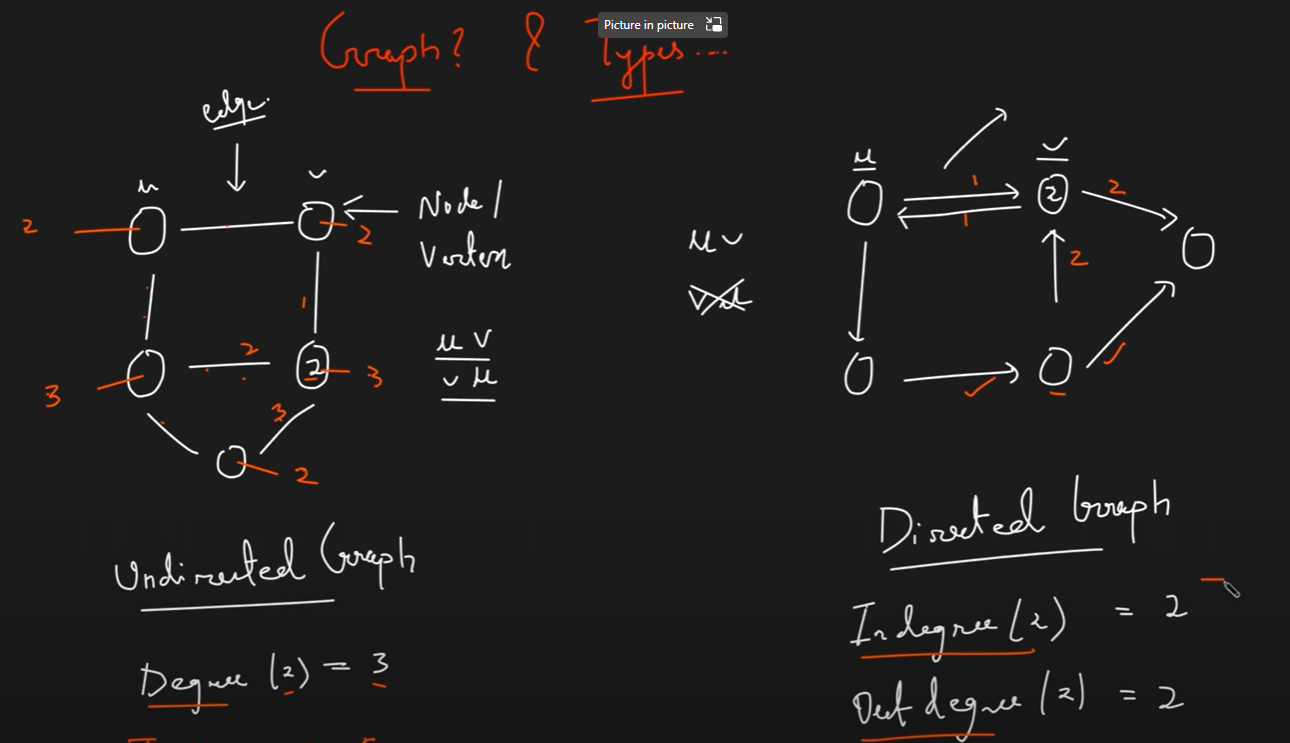
We should use adjacency list

DEGREE

In undirected graph degree is the number of edges connecting to a vertex

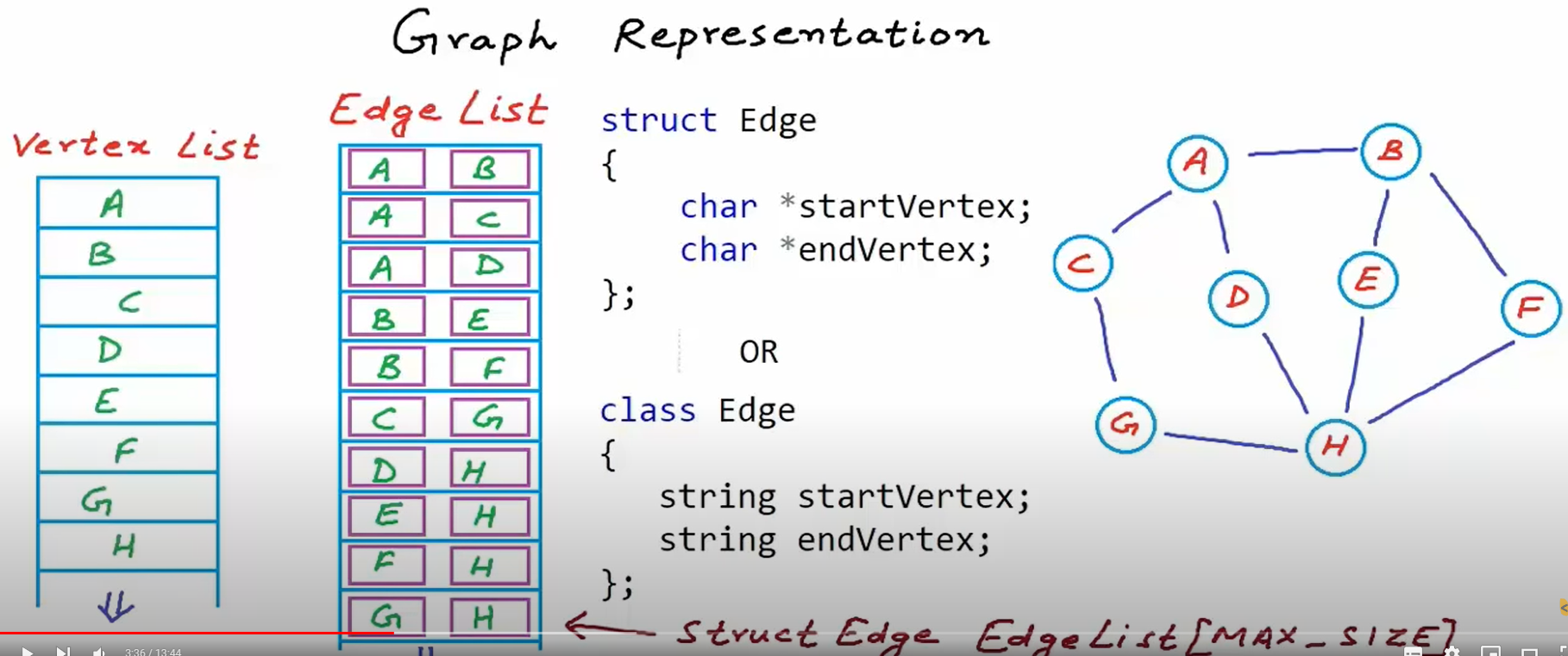
In directed graph indegree is the number of edges coming towards a vertex

In directed graph outdegree is the number of edges going outward from a vertex

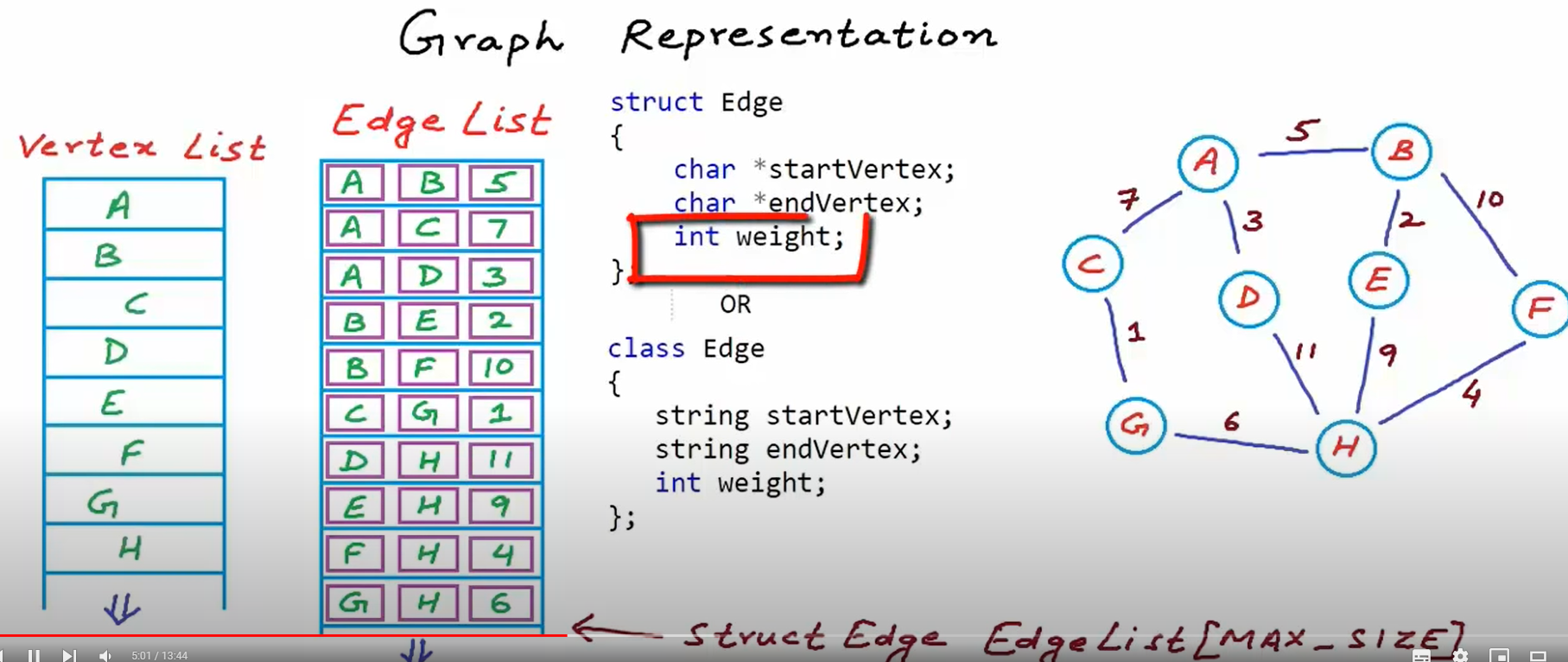


**A 🡪 EDGE LIST**

**Unweighted edges**

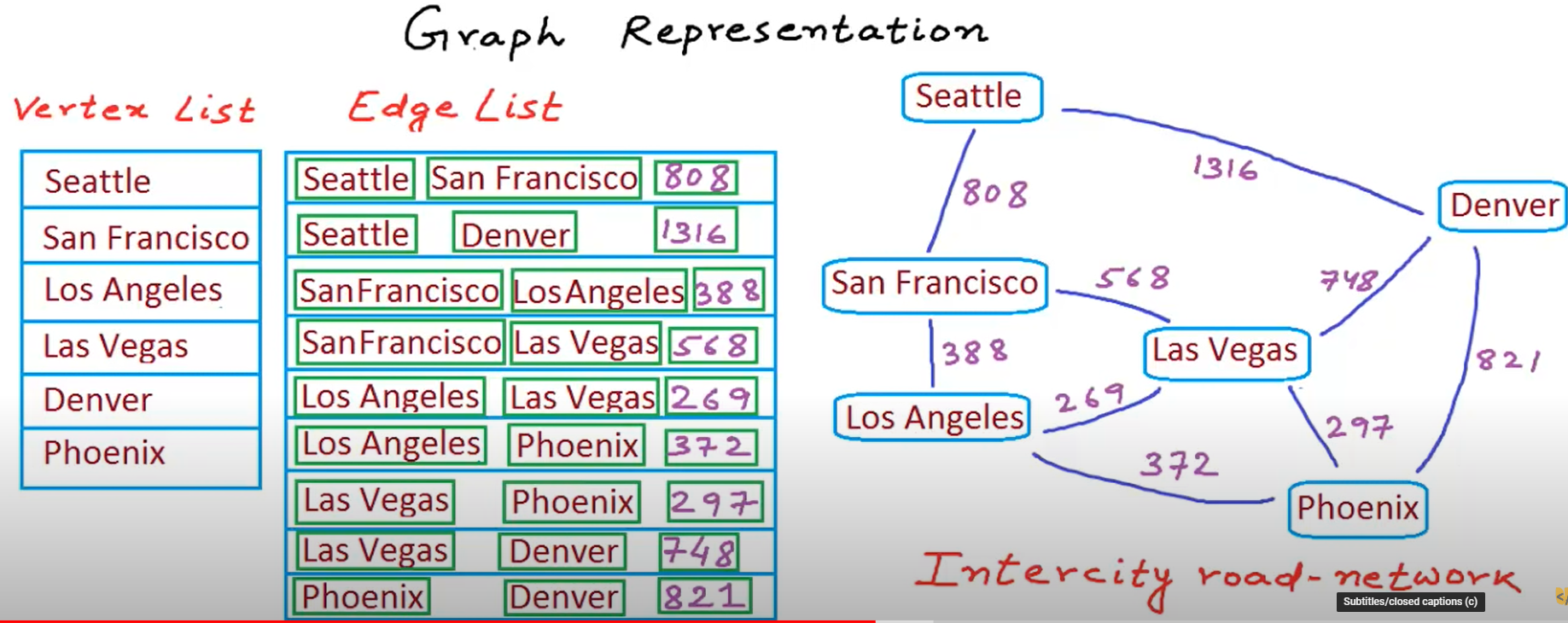


**Weighted edge**



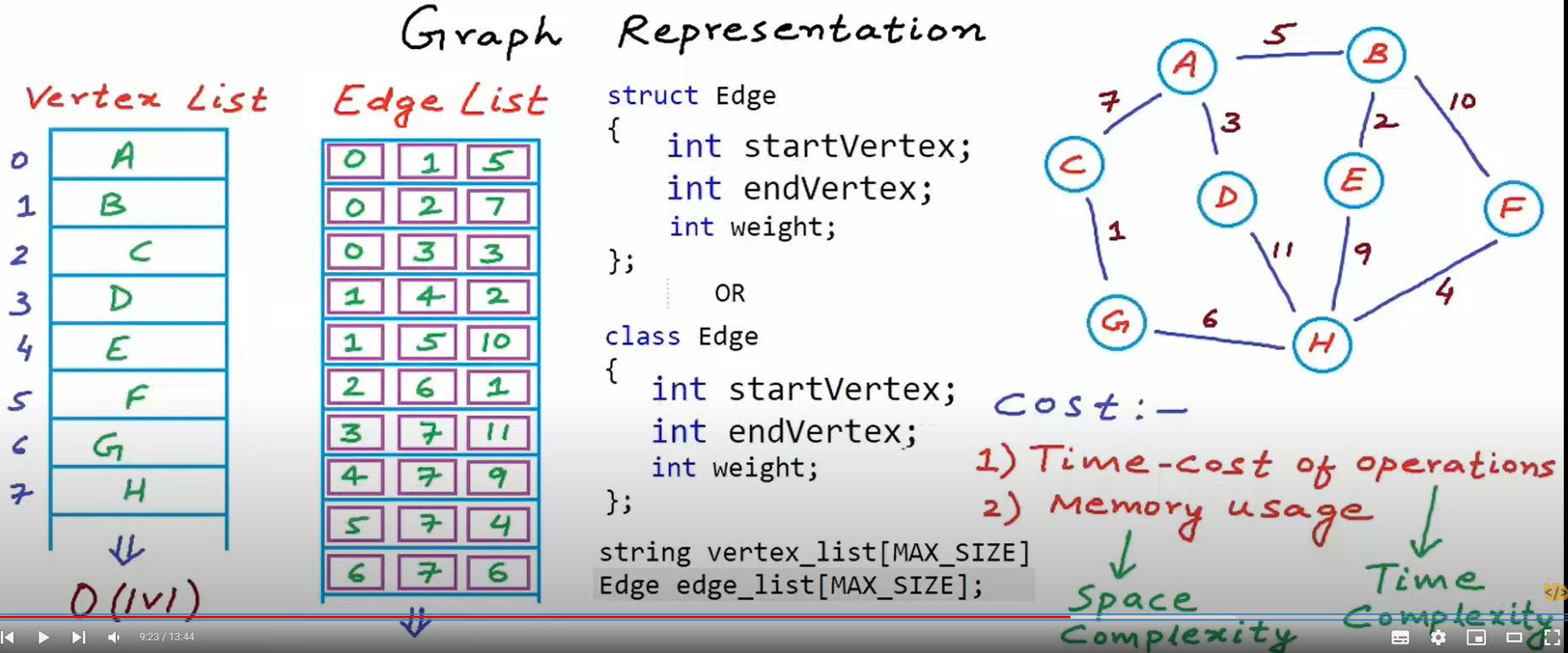
**Drawbacks**

1. Since edge list stores the name of the Vertex (String type) If the name is big enough then the memory taken will be a lot .Example in below ss



**Solution:** In order to overcome this issue of memory consumption we can just refer the vertex address in vertex list. Example in below ss

As you can see the edge list first 2 elements contains the index of vertex in vertex list and not the actual value of the vertex



**Time Complexity:**

1. Finding all nodes adjacent to a given node (vertex)

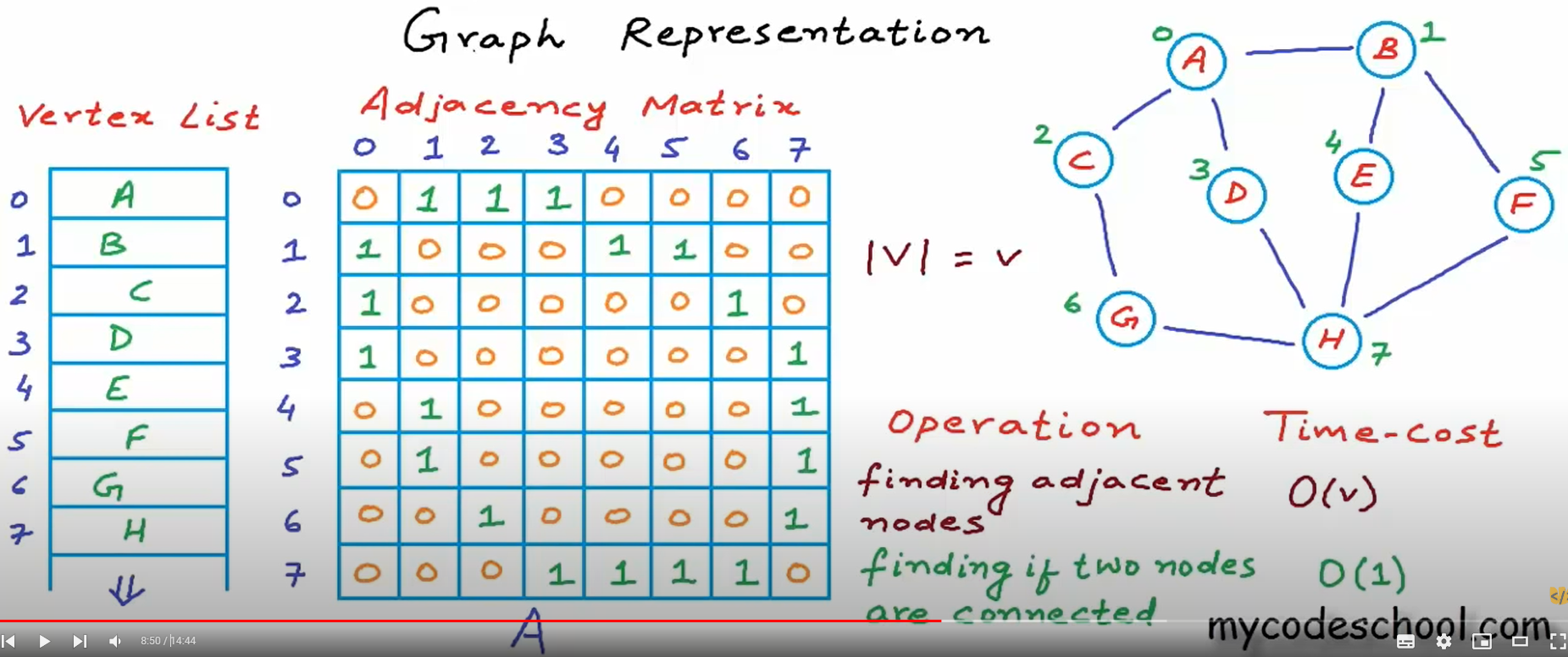
O(e) 🡪 which means that we will have to traverse the entire list of edges in worst case scenario to get nodes

1. Check if given nodes are connected

Again O(e) time will be required which in case of a graph is a lot

**B 🡪 ADJACENCY MATRIC**

**For unweighted graph**



**Weighted graphs**

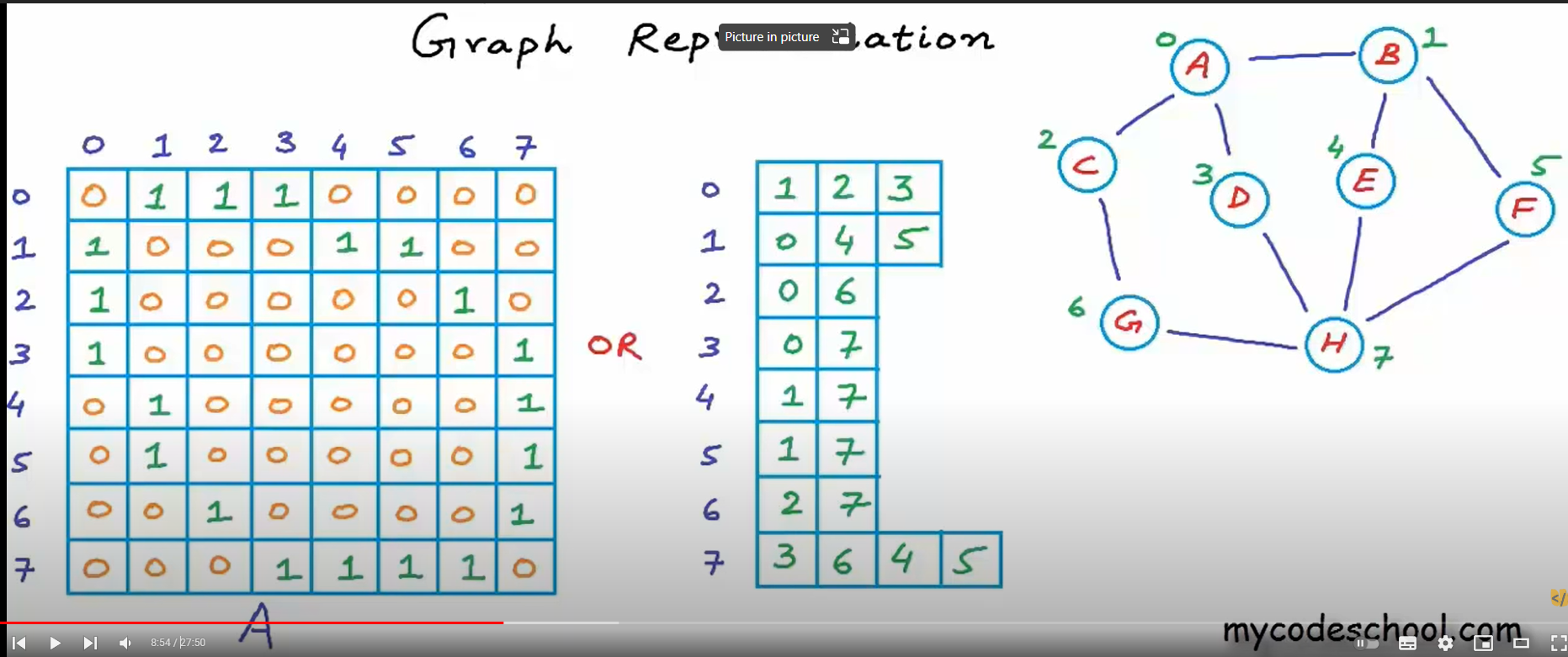


**Drawbacks:**

In this approach memory used is more because we are also storing the information of vertexs that are not connected that is 0 in first screenshot and infinite in the second screenshot

This drawback can be overcomed by using adjacency list

**ADJACENCY LIST**



Instead of storing the 0’s as well we just store the list of vertexes in connected to each vertex like in this case A(index 0) is connected to B(index 1),C(index 2) and D(index 3)

* We use List<List<Integer>> to represent adjacency list
* In case it’s a weighted graph we can use List<List<Pair<Integer,Integer>>

DISCONNECTED GRAPH

This not called 3 different graphs but it is called as 1 disconnected graph with 3 components

