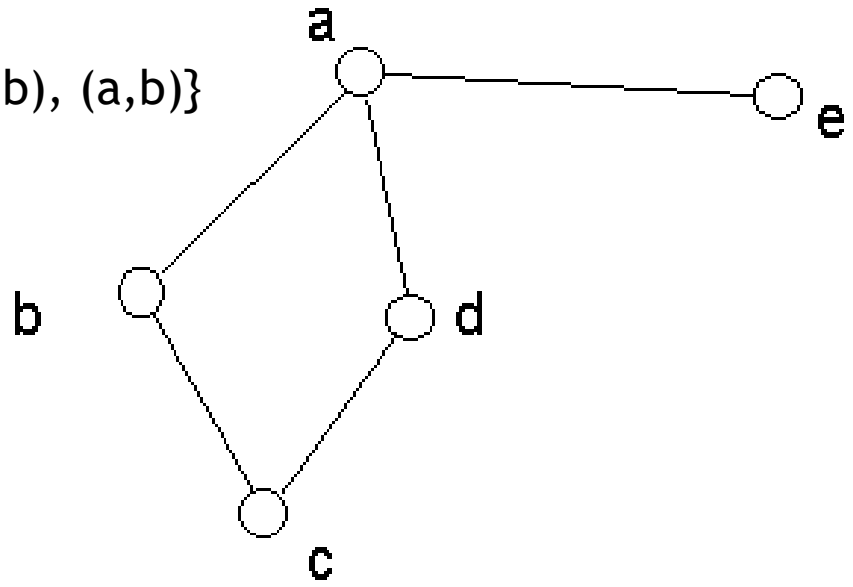
Intro To Graphs

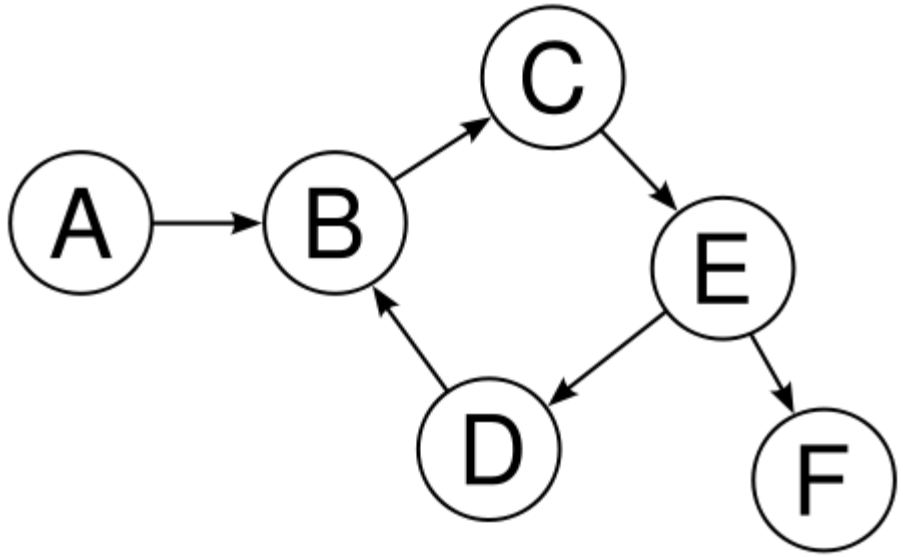
CodeISM 2024

What is a graph?

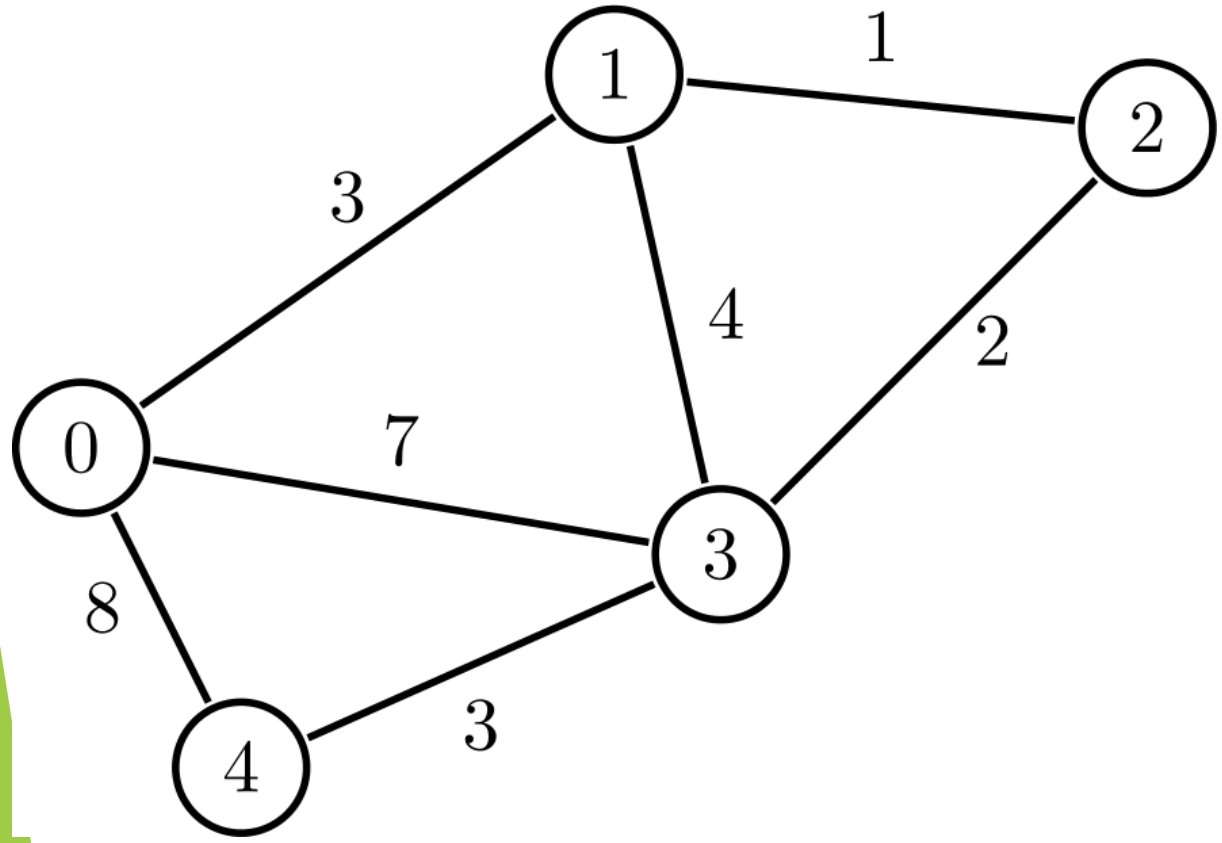
Graph = Set of vertices + Set of edges
In given graph:
Vertices = {a, b, c, d, e}
Edges = {(a,e), (a, d), (d,c), (c,b), (a,b)}



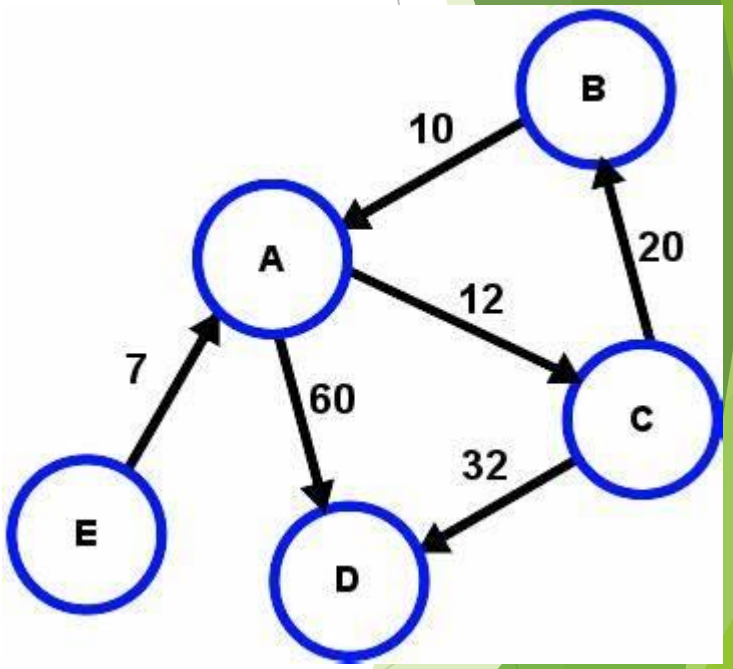
Directed Graph



Weighted Graphs

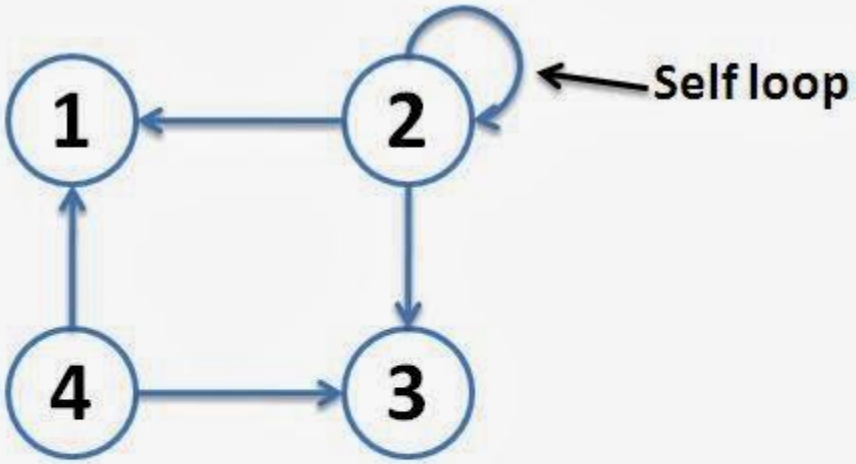


Undirected Weighted Graph

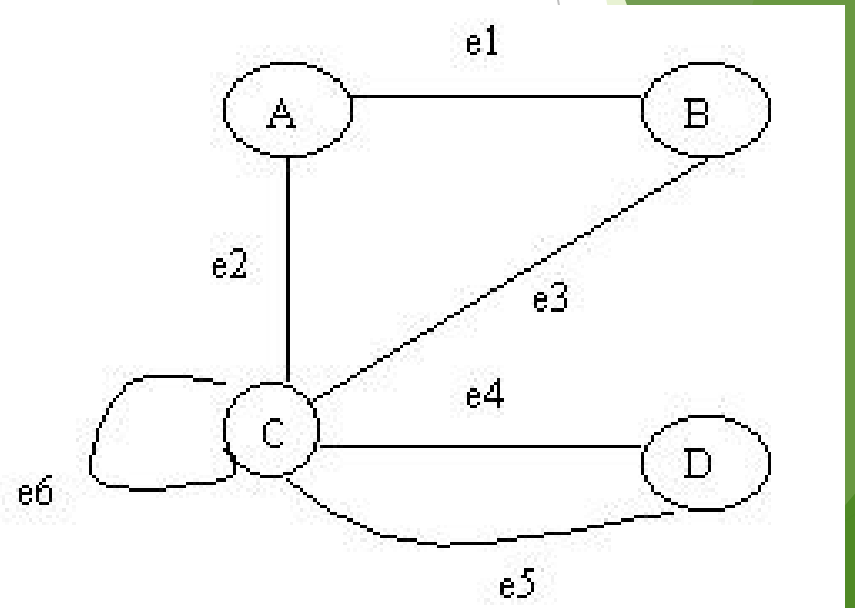


Directed Weighted Graph

Self Loops and Parallel Edges



Graph with self loop

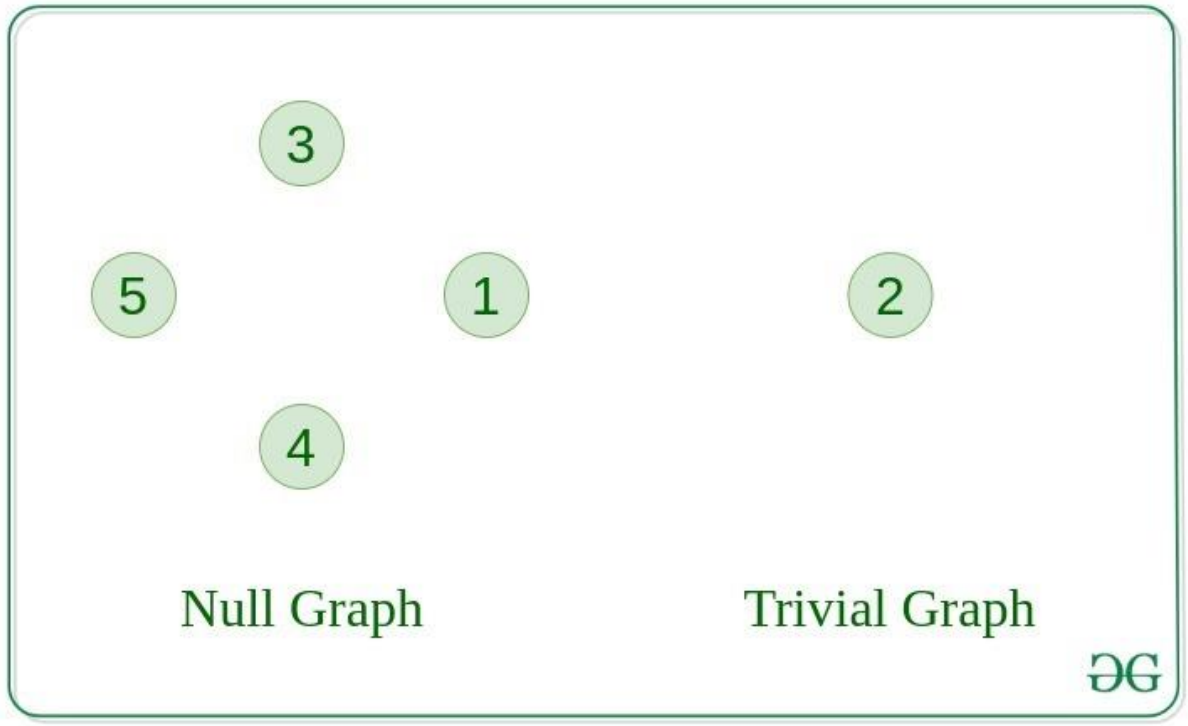


e4 and e5 are parallel edges

Simple Graph: Graph with **no self loops** and **no parallel edges.**

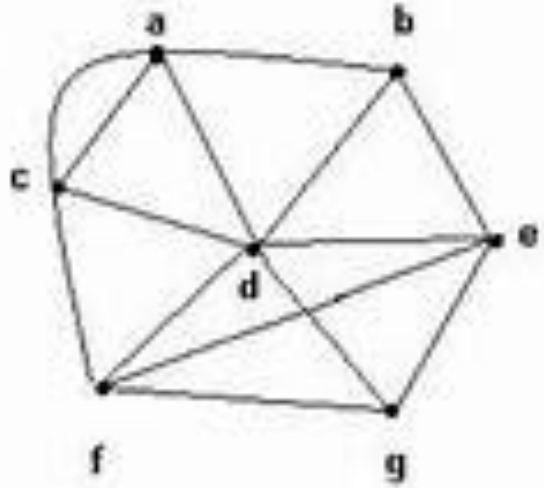
Null and Trivial Graph

Null Graph: Has only vertices, **no edges**
Trivial Graph: Has a **single vertex** and no edges (Null graph with single vertex)



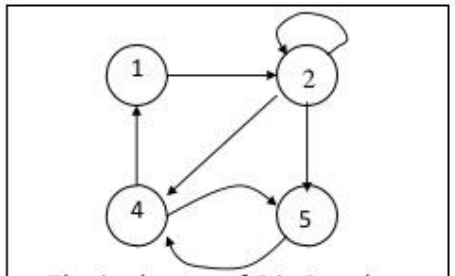
Degree of a vertex

- The *degree* of a vertex v, denoted by $\delta(v)$, is the number of edges incident on v
- Example:
  - $\delta(a) = 4$, $\delta(b) = 3$,
  - $\delta(c) = 4$, $\delta(d) = 6$,
  - $\delta(e) = 4$, $\delta(f) = 4$,
  - $\delta(g) = 3$.



Degree (Directed Graphs)

- In degree: Number of edges entering a node
- Out degree: Number of edges leaving a node
- Degree = Indegree + Outdegree



The in degree of 2 is 2 and
the out degree of 2 is 3.

Graph Representation: Adjacency Matrix

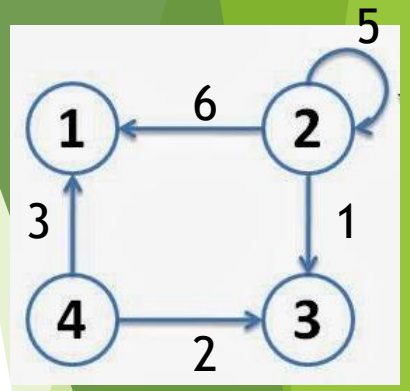
	1	2	3	4
1	i	i	i	i
2	6	5	1	i
3	i	i	i	i
4	3	i	2	i

Weighted Directed

	1	2	3	4
1	0	0	0	0
2	1	1	1	0
3	0	0	0	0
4	1	0	1	0

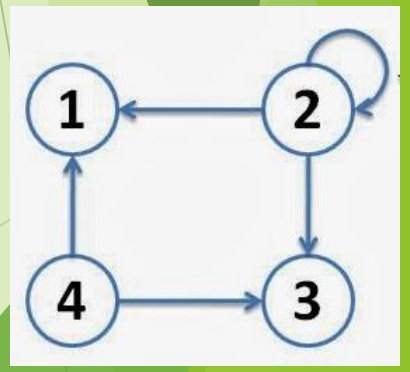
Unweighted Directed

Graphs



Weighted Version

Notice the symmetricity in **undirected graphs**



Unweighted Version

	1	2	3	4
1	i	6	i	3
2	6	5	1	i
3	i	1	i	2
4	3	i	2	i

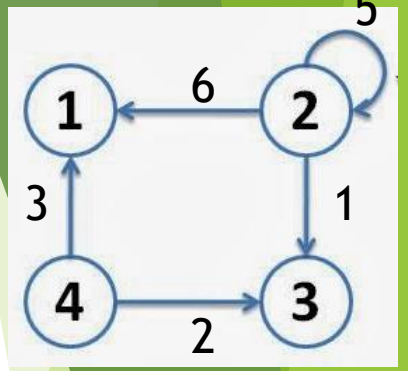
Weighted Undirected

	1	2	3	4
1	0	1	0	1
2	1	1	1	0
3	0	1	0	1
4	1	0	1	0

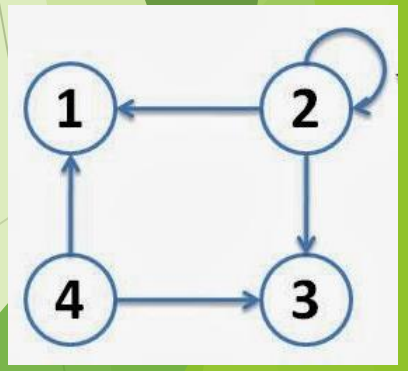
Unweighted Undirected

Graph Representation: Adjacency List

Graphs



Weighted Version



Unweighted Version

1 -> empty
2 -> (1, 6), (2, 5), (3, 1)
3 -> empty
4 -> (1, 3), (3, 2)

Weighted Directed

1 -> empty
2 -> 1, 2, 3
3 -> empty
4 -> 1, 3

Unweighted Directed

1 -> (2, 6), (4, 3)
2 -> (1, 6), (2, 5), (3, 1)
3 -> (2, 1), (4, 2)
4 -> (1, 3), (3, 2)

Weighted Undirected

1 -> 2, 4
2 -> 1, 2, 3
3 -> 2, 4
4 -> 1, 3

Unweighted Undirected

Efficiency Comparison Parameters

- Space Concern: How much space is taken up by a certain representation
- Time Concern (Lookup Time) : How much time is required to know whether edge exists between two given vertices.

Comparison Of Representations

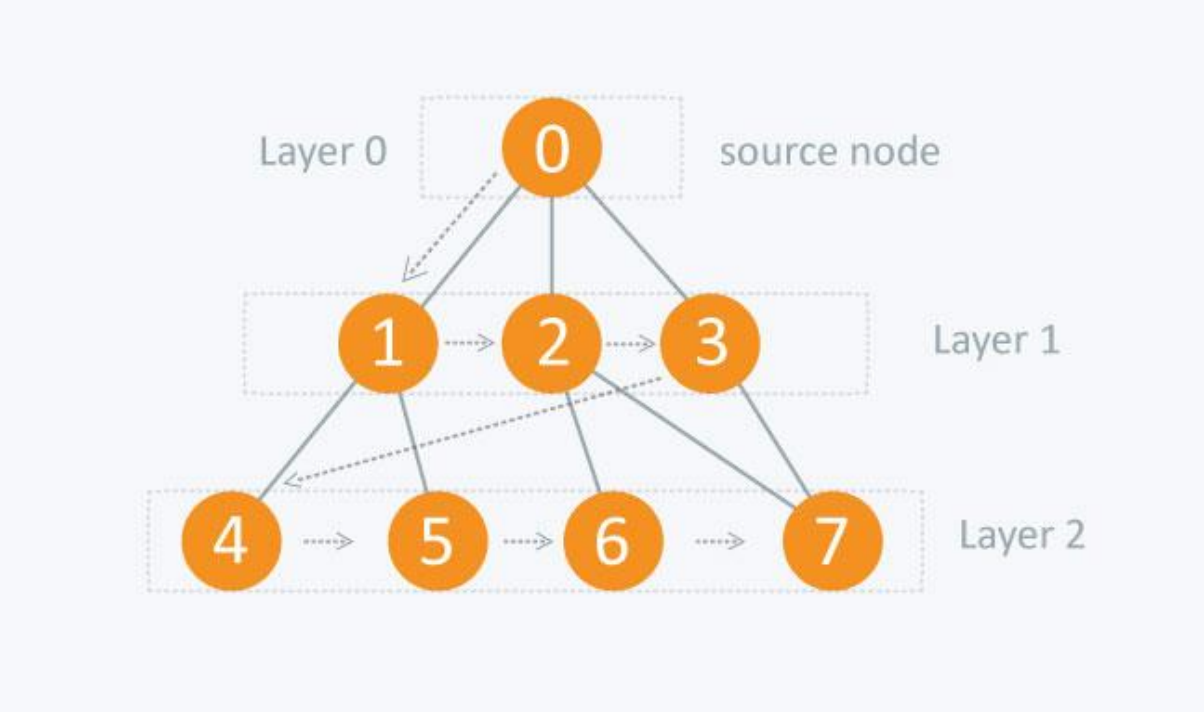
Adjacency Matrix

- Space: $O(V^2)$
- Lookup Time: O(1)
- Unweighted Graphs: bool[][]
- Weighted Graphs: int[][]

Adjacency List

- Space -> O(E)
- Time -> O(V)
- Unweighted: vector<int>[]
- Weighted: vector<pii>[]

Graph Traversal: Breadth First Search




```
//newly introduced vertices at
//every instance
//time = 0 -> [0]
//time = 1 -> [1, 2, 3]
//time = 2 -> [4, 5, 6, 7]
```

Problems and Codes: Doc Link