Introduction

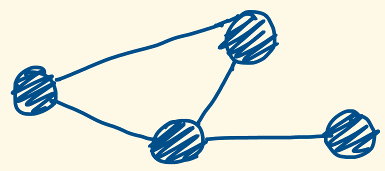
Wednesday, October 25, 2023   5:17 AM

What is graph theory?

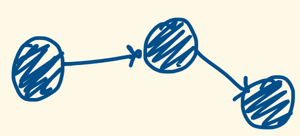
Applications of graphs in real world situations.
- Social network
- Map, etc.

Types of graphs:

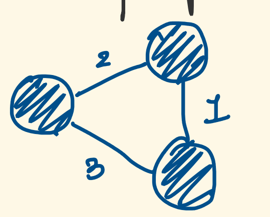
1. Undirected graphs: edges have no orientations.
   eg. Nodes as cities & edges as roads.



2. Directed graphs: edges have orientations
   eg. people buying gifts for each other.



3. Weighted graph: edges hv weights

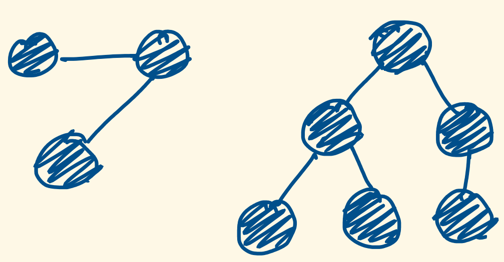


4. Trees: undirected graphs with no cycles

A tree has no cycles!

- A tree with N nodes is a connected graph with N-1 edges.

A path between every node in a graph.

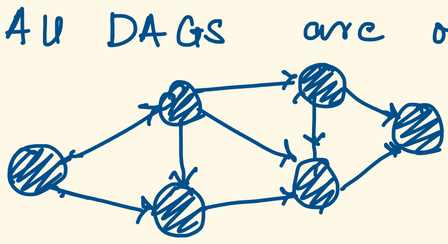


↳ A Rooted tree.
- A tree with every node pointing towards the root. ← Aborescence (out-tree)
  or pointing away from the root. ← Anti-aborescence (in-tree)

5. Directed Acyclic Graphs (DAGs)

- Common for representind dependencies.
  eg. task scheduling, class prerequisites, etc.

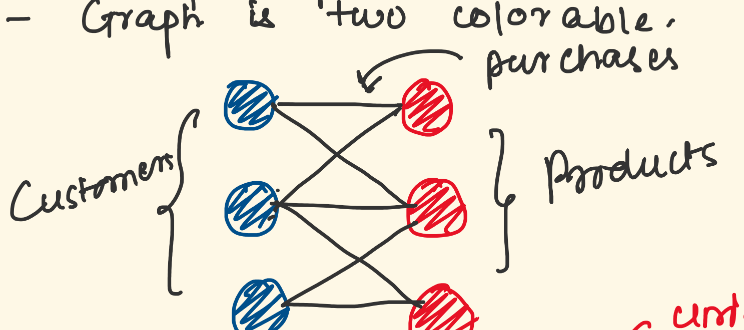
  • All DAGs are out trees but not the other way round.



6. Bipartite Graph

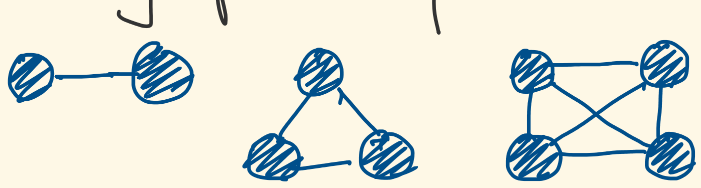
- Vertices can be split into two groups (U & V)
- Every edge connects between U & V.
- Graph is two colorable.

They have nothing in common.
No edges between nodes of the same group



7. Complete graph: Edge between every pair of nodes.

unique edge



How to represent graphs?

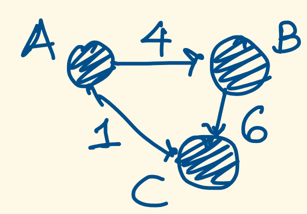
1. Adjacency matrix

- m[i][j] → cost of i to j
- Space efficient for dense graphs
- lookup time is O(1)
- Requires V^2 space & time.

2. Adjacency list.

- Map of lists / list of lists.

  A → [(B,4), (C,1)]
  B → [(C,6)]



- Space efficient for sparse graphs.
- Less efficient for dense graphs.
- Lookup time → O(E)
  ↳ Edges