IIT ROORKEE

FREE ONLINE EDUCATION swayam शिक्षित भारत, उन्नत भारत

NPTEL ONLINE CERTIFICATION COURSE

VLSI Physical Design with Timing Analysis

Lecture – 4: Graphs for Physical Design

Bishnu Prasad Das

Department of Electronics and Communication Engineering

Contents

Introduction

- Graphs are fundamental to VLSI physical design.
- They provide a versatile and powerful tool for
  - **representing, analyzing,** and **optimizing** complex electronic circuits.

Introduction

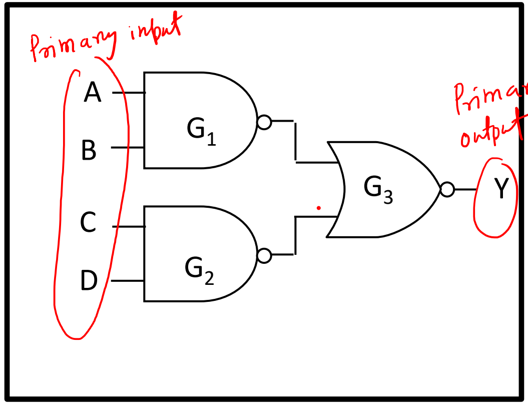
Some of the applications of Graphs in Physical Design are:

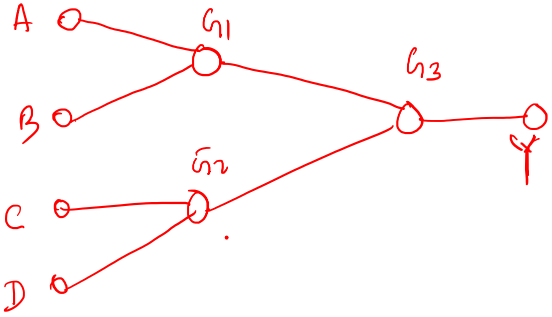
- **Netlist Representation**: Graphs represent connections between electronic components in a VLSI circuit
- **Floor Planning**: Graphs depict relationships and constraints between blocks.

Introduction

- **Routing**: Graphs help routing algorithms for optimal wire connections.
- **Critical Path Analysis**: Graph-based algorithms help identify the longest (critical paths) paths in a VLSI design.

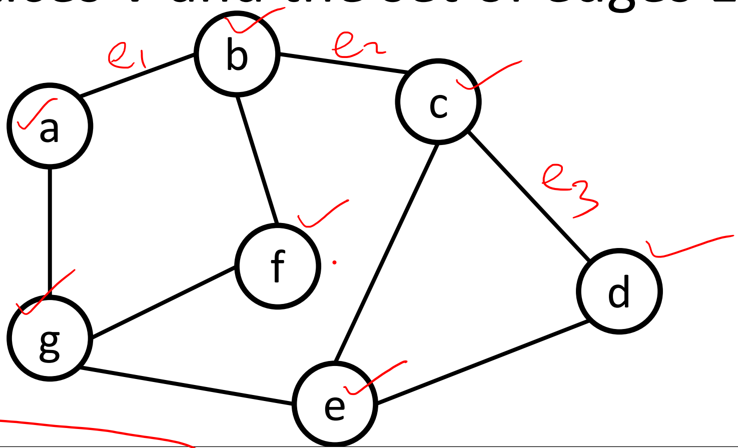
Graph Representation of the Logic Circuit





Terminology

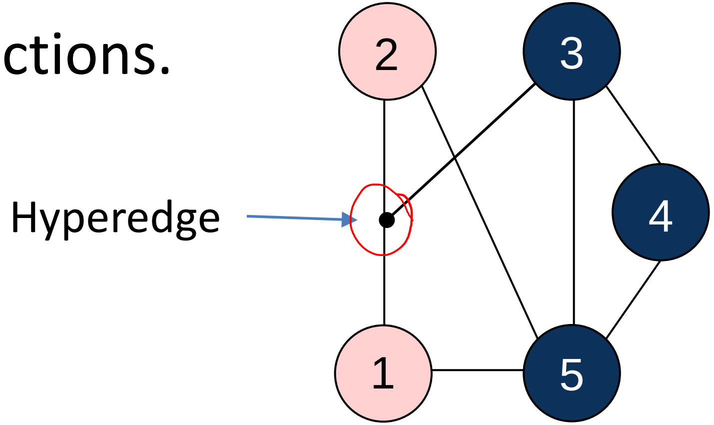
- **Graph**: A graph G(V, E) is made up of two sets – the set of nodes or vertices V and the set of edges E.



V = {a, b, c, d, e, f, g}
E = {(a, b), (b, c), (c, d), (d, e), (e, g), (g, f), (a, g), (b, f), (c, e)}
e1

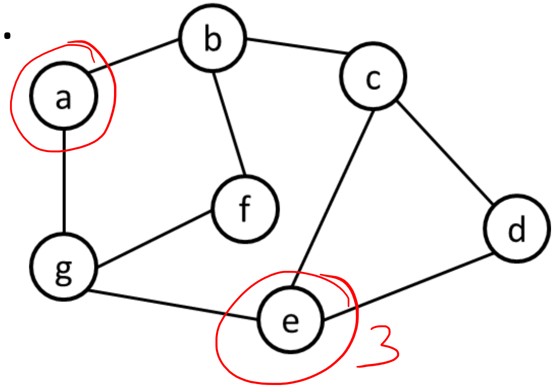
Terminology

- **Hypergraph**: Consists of nodes and hyperedges, with each hyperedge being a subset of two or more nodes.
- Hyperedges are commonly used to represent multi-pin nets or multi-point connections.



Terminology

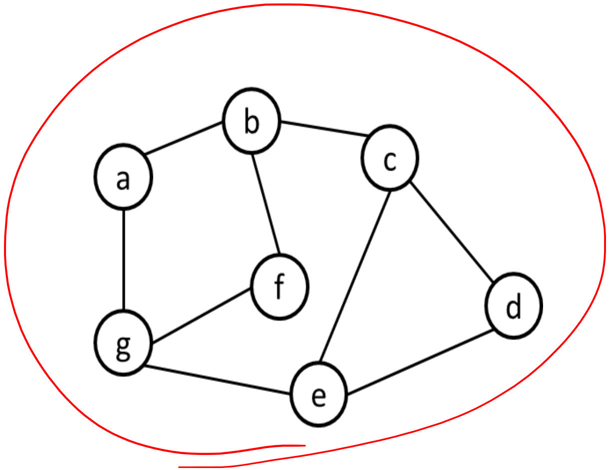
- **Degree of the node**: Number of its incident edges.
- **Path:** A path between two nodes is an ordered sequence of edges from the start node to the end node.
- **Loop:** A cycle (loop) is a closed path that starts and ends at the same node.



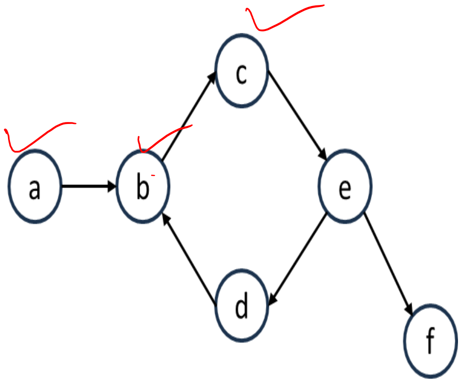
Terminology

- **Undirected Graph:** Represents only unordered node relations with no directed edges.
- **Directed graph:** A graph in which the direction of the edge denotes a specific ordered relation between two nodes.

Terminology



Undirected Graph

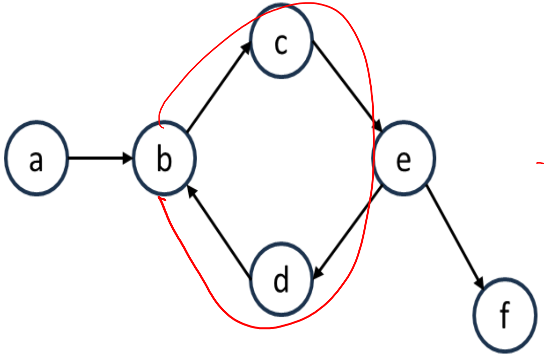


Directed Graph

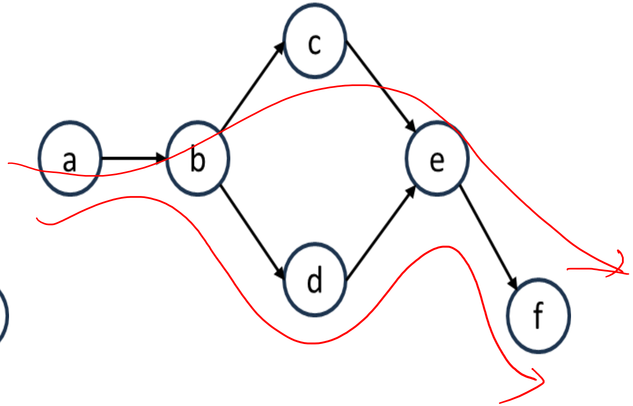
Terminology

- **Cyclic Graph:** A directed graph has at least one directed cycle. Otherwise, it is an **Acyclic Graph**.
  - The design data used in several EDA algorithms is represented in **Directed Acyclic Graphs**(DAG).

Terminology



Cyclic Graph

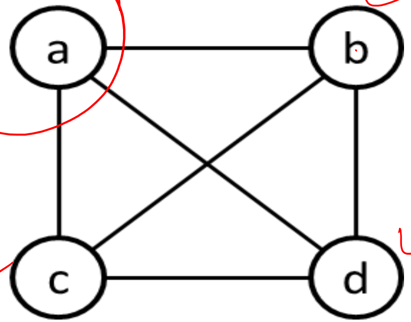


Acyclic Graph

Terminology

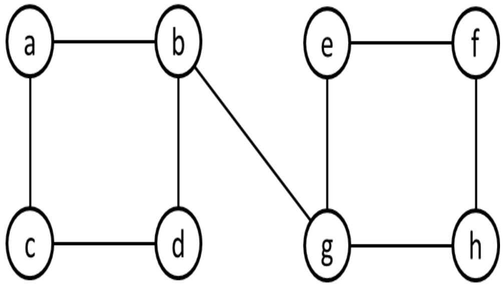
- **Complete Graph:** A graph in which an edge connects each node to every other node.

- Contains $^{n}C_2 = \dfrac{n(n-1)}{2}$ edges

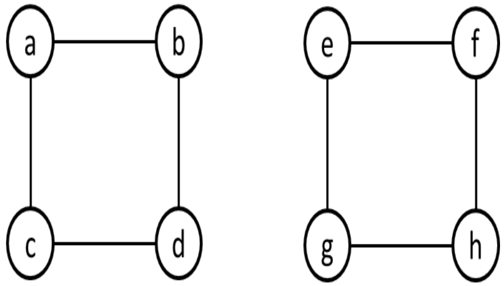


Terminology

- **Connected Graph:** A graph with at least one path between each pair of nodes.



Connected Graph

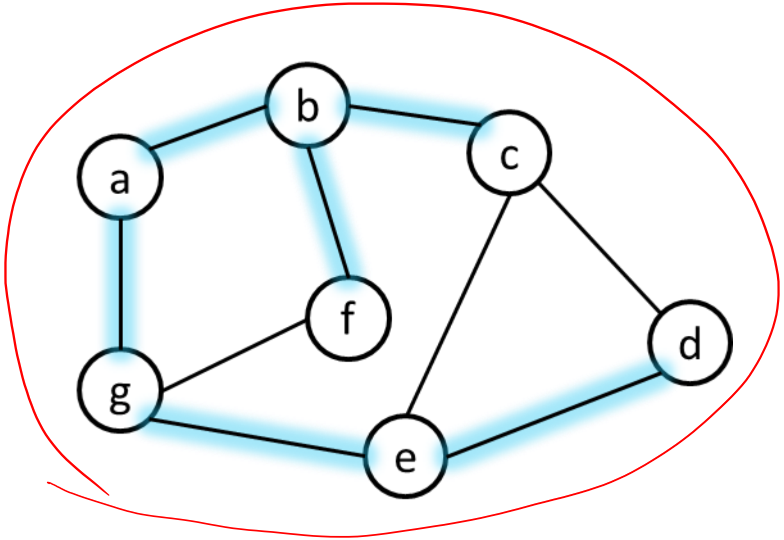


Unconnected Graph

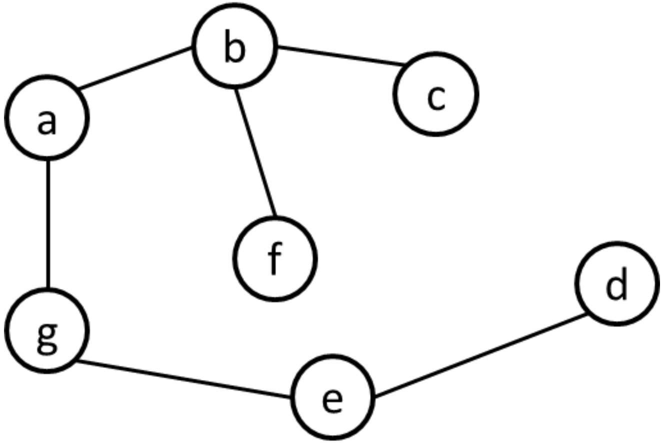
Terminology

- **Tree:** A graph with n nodes connected by n – 1 edges.
  - A tree does not contain any cycles.
  - A tree is a maximal acyclic graph,
    - which means that adding an edge between two nodes that are not connected would create a cycle.
  - Each pair of nodes is connected by exactly one edge.
  - There are two types of trees: **undirected and directed.**

Terminology



Graph



Tree

Terminology

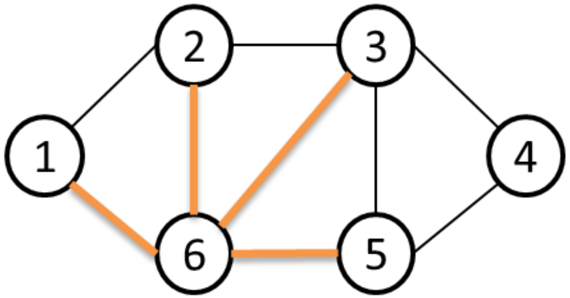
- **Spanning Tree:** A connected, acyclic subgraph G' contained within G(V, E) that includes (spans) every node v ϵ V.
- **Minimum spanning tree (MST):** A spanning tree with the smallest possible sum of edge costs (i.e., edge lengths).

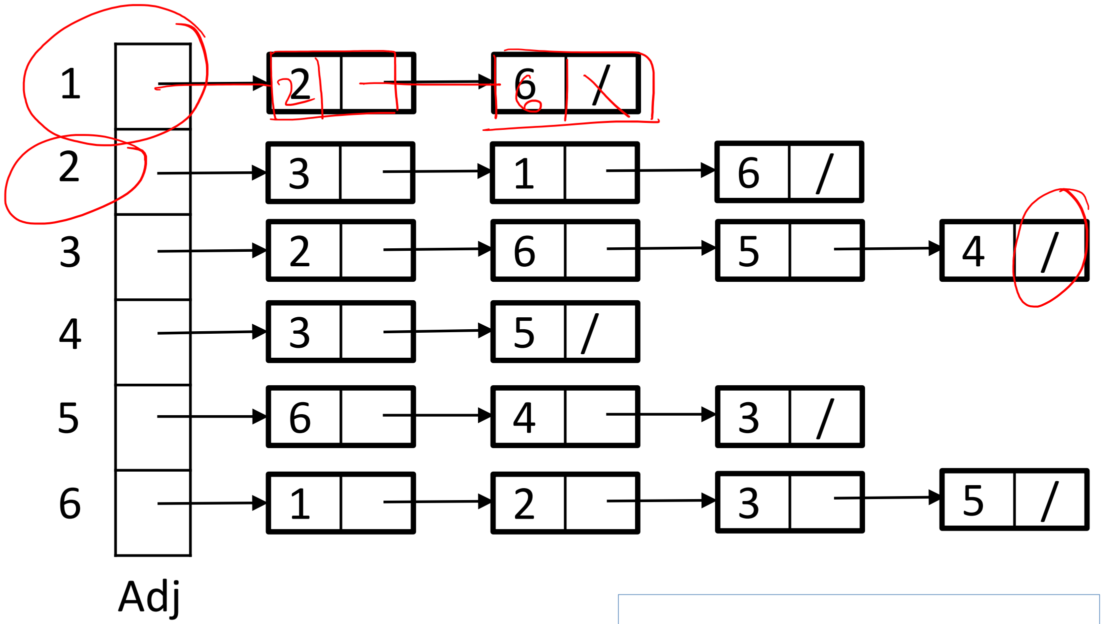
Representation of Graphs

- There are two standard ways of representing a graph:
  - Adjacency Lists
    - For Sparse graphs – for which $|E|$ is much less than $|V|^2$, Adjacency list representation is preferred as they provide compact representation.
  - Adjacency Matrices
    - For dense graphs – $|E|$ is close to the $|V|^2$, Adjacency Matrix representation is preferred.

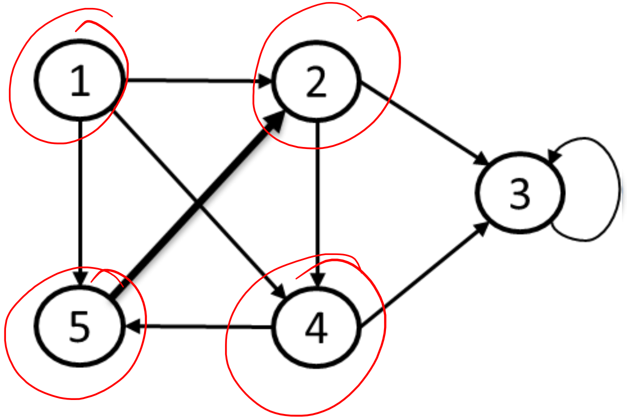
Adjacency List Representation of Graphs

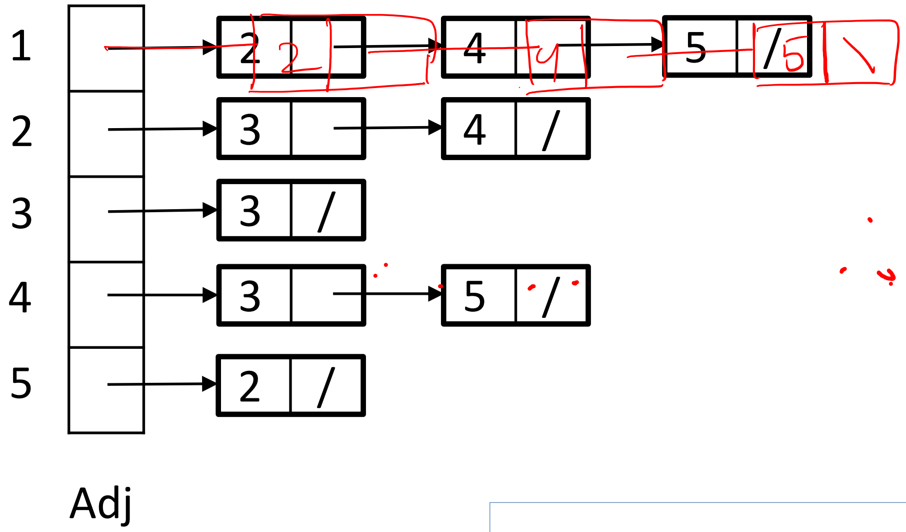
- The **adjacency-list representation** of a graph $G = (V, E)$ consists of an array *Adj* of $|V|$ lists, one for each vertex in V.
- For each $u \in V$, the adjacency list *Adj[u]* contains all the vertices v such that there is an edge $(u, v) \in E$.





Adjacency List Representation of the Undirected Graph

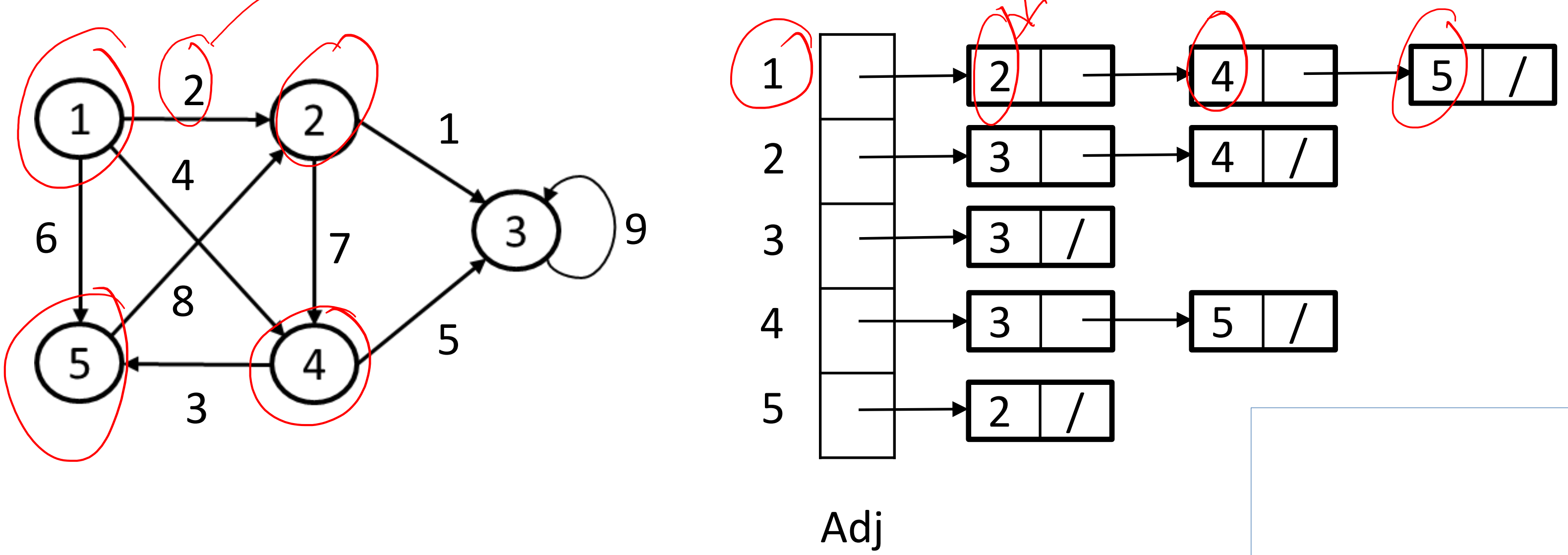




Adjacency List Representation of the Directed Graph

Representation of Graphs

- For example, let G = (V, E) be a weighted graph with weight function *w*.
  - The weight w(u, v) of the edge (*u, v*) ϵ E can be stored with vertex *v* in *u*'s adjacency list.



Adjacency List Representation of the weighted directed Graph

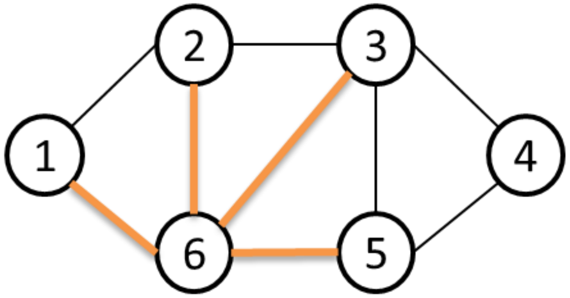
Adjacency Matrix Representation of Graphs

- The adjacency matrix representation of a graph G consists of a $|V| \times |V|$ matrix $A = (a_{ij})$ such that

$G(V,E)$

$$a_{ij} = \begin{cases} 1 & \text{If } (i, j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

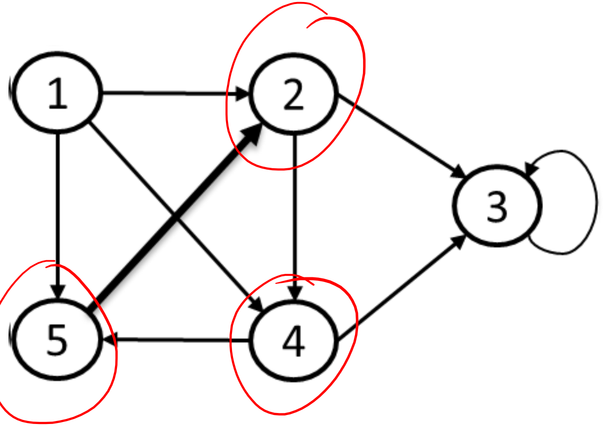
$|V| \times |V|$



6 X 6

	1	2	3	4	5	6
1	0	1	0	0	0	1
2	1	0	1	0	0	1
3	0	1	0	1	1	1
4	0	0	1	0	1	0
5	0	0	1	1	0	1
6	1	1	1	0	1	0

Adjacency Matrix Representation of the Undirected Graph



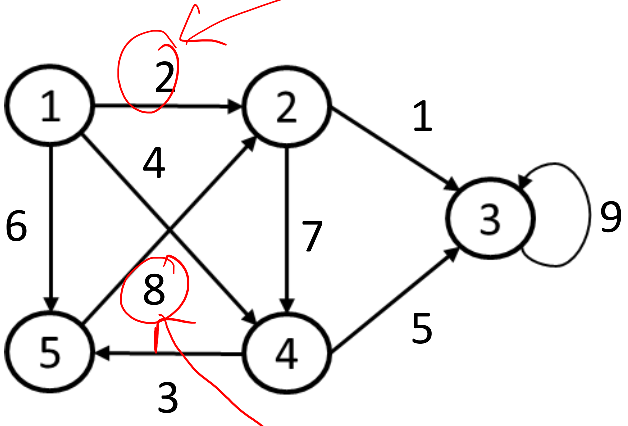
5 X 5

	1	2	3	4	5
1	0	1	0	1	1
2	0	0	1	1	0
3	0	0	1	0	0
4	0	0	1	0	1
5	0	1	0	0	0

Adjacency Matrix Representation of the Directed Graph

Adjacency Matrix Representation of Graphs

- For example, if G = (V, E) is a weighted graph with edge-weight function w.
  - The weight w(u, v) of the edge (u, v) ϵ E can be stored as the entry in row u and column v of the adjacency matrix.



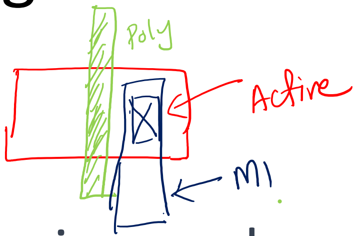
	1	2	3	4	5
1	0	2	0	4	6
2	0	0	1	7	0
3	0	0	9	0	0
4	0	0	5	0	3
5	0	8	0	0	0

Adjacency Matrix Representation of the weighted directed Graph

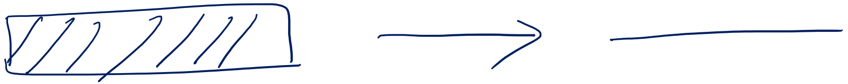
Adjacency Matrix vs Adjacency List

Adjacency List	Adjacency Matrix
The amount of memory it requires is $\Theta(V+E)$.	The adjacency matrix of a graph requires $\Theta(V^2)$ memory
Finding each edge in the graph takes $\Theta(V+E)$ time.	Because finding each edge in the graph requires examining the entire adjacency matrix, doing so takes $\Theta(V^2)$ time.

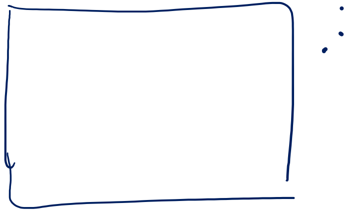
Classes of Graphs in Physical Design



- Layouts consist of collections of rectangles.
- In *routing problems*, rectangles represent routing wires and are often thin and long.



- In *placement and compaction problems*, rectangles represent circuit blocks.

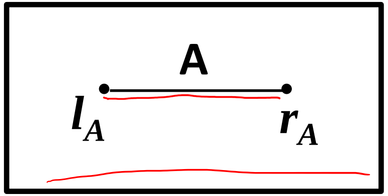


Classes of Graphs in Physical Design

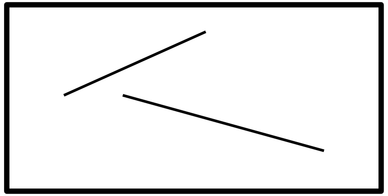
- To optimize the arrangement of lines or rectangles in two or three dimensions, various graphs are defined to represent their relationships and adjacencies.
  - Graphs Related to a Set of Lines
  - Graphs Related to a Set of Rectangles

Graphs Related to a Set of Lines

- Lines can be classified into two types:



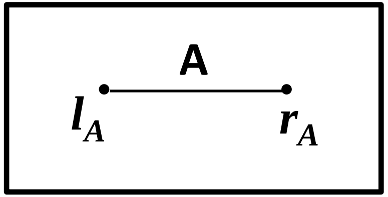
Aligned to the axis



Not aligned to the axis

Graphs Related to a Set of Lines

- **Interval**: An interval is represented by its left and right endpoints, denoted by l_i and $r_{i,}$ respectively for lines aligned to axes.



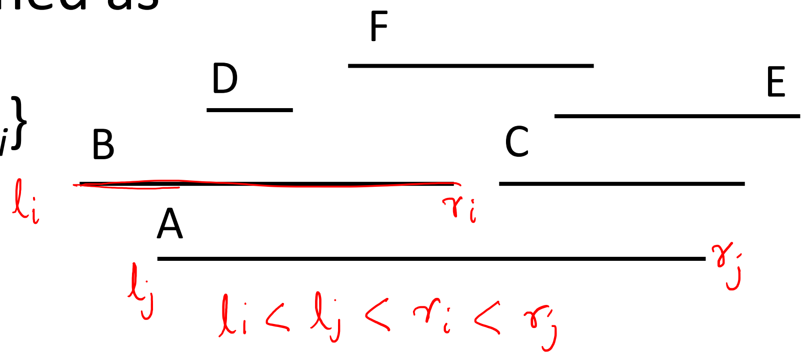
Interval A: (l_A, r_A)

Graphs Related to a Set of Lines

- Given a set of intervals $I = \{I_1, I_2, .., I_n\}$, three graphs are defined based on their different relationships.
  - Overlap graph
  - Containment graph
  - Interval graph
- Overlap, containment, and interval graphs arise in many routing problems.

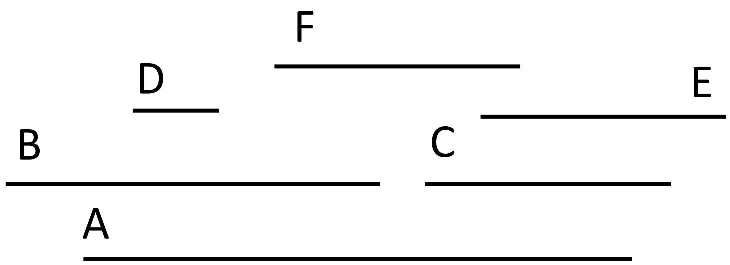
Overlap graph

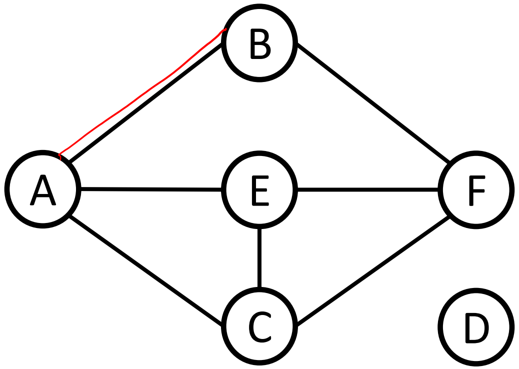
- **Overlap graph** $G_O = (V, E_O)$, is defined as
  - $V = \{v_i \mid v_i$ represents interval $I_i\}$
  - $E_O = \{(v_i, v_j) \mid l_i < l_j < r_i < r_j\}$



- An edge is defined between v_i and v_j if interval I_i overlaps with I_j but does not completely contain or reside within I_j.

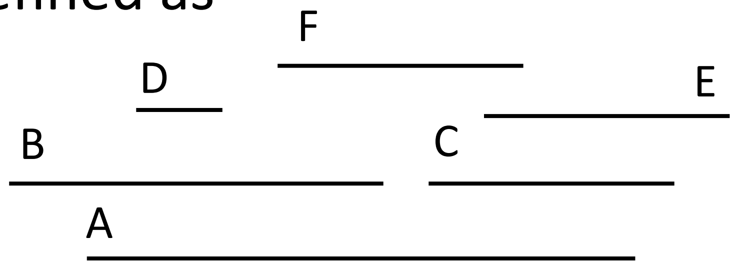
Overlap graph



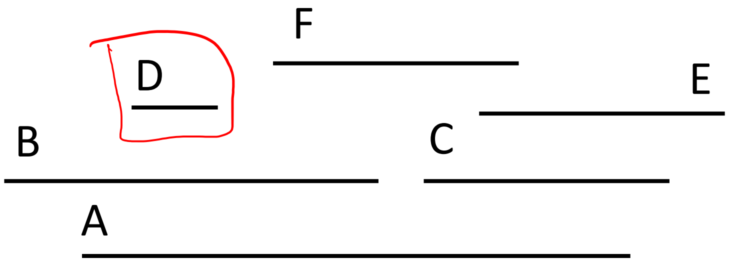


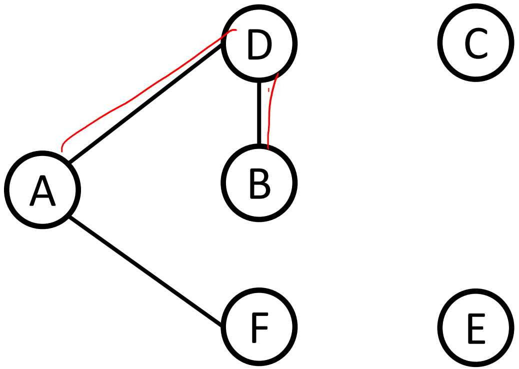
Containment graph

- **Containment graph** $G_C = (V, E_C)$, is defined as
  - $V = \{v_i \mid v_i \text{ represents interval } I_i\}$
  - $E_C = \{(v_i, v_j) \mid l_i < l_j, r_i > r_j\}$
- An edge is defined between v_i and v_j if interval I_i completely contains I_j.



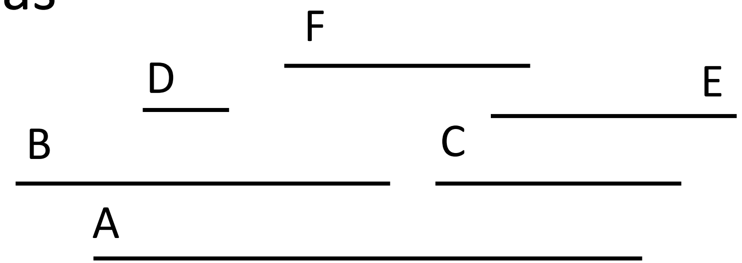
Containment graph



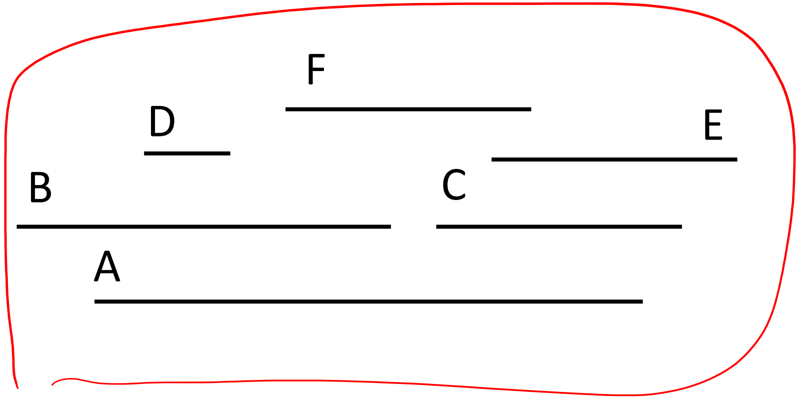


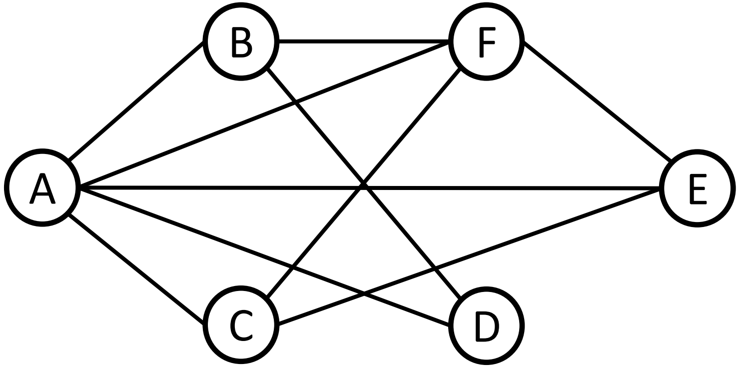
Interval graph

- **Interval graph** $G_I = (V, E_I)$, is defined as
  - $V = \{v_i \mid v_i$ represents interval $I_i\}$
  - $E_I = E_O \cup E_C$
- An edge is defined between v_i and v_j if interval I_i has a non-empty intersection with I_j.



Interval graph



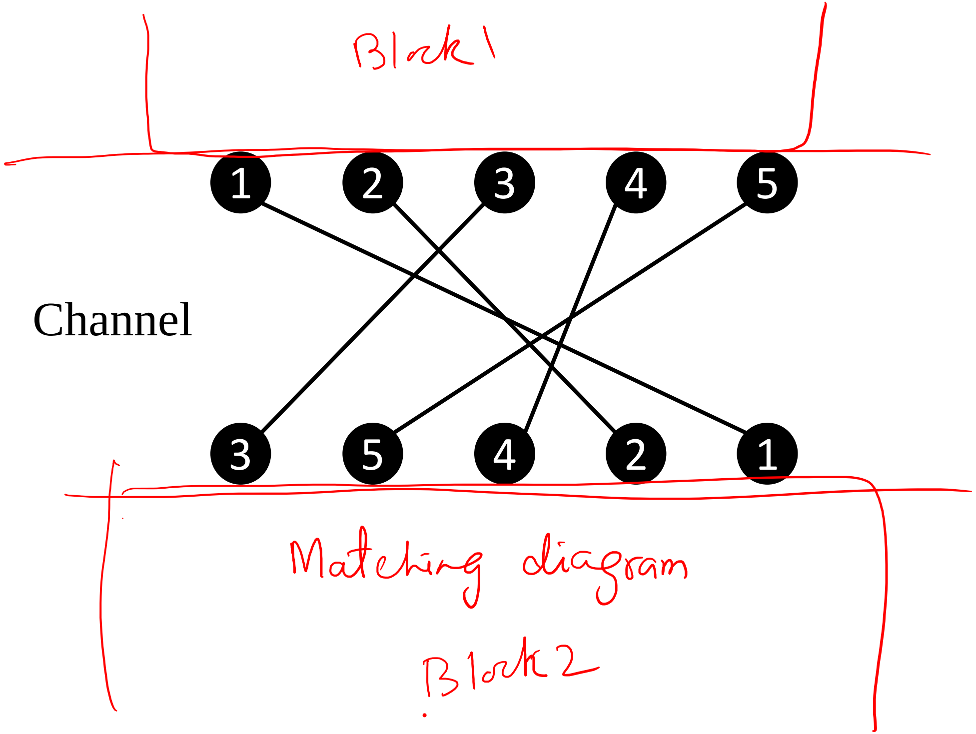


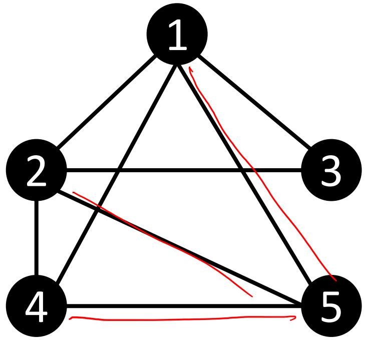
$E_I = E_0 \cup E_c$

Permutation Graph

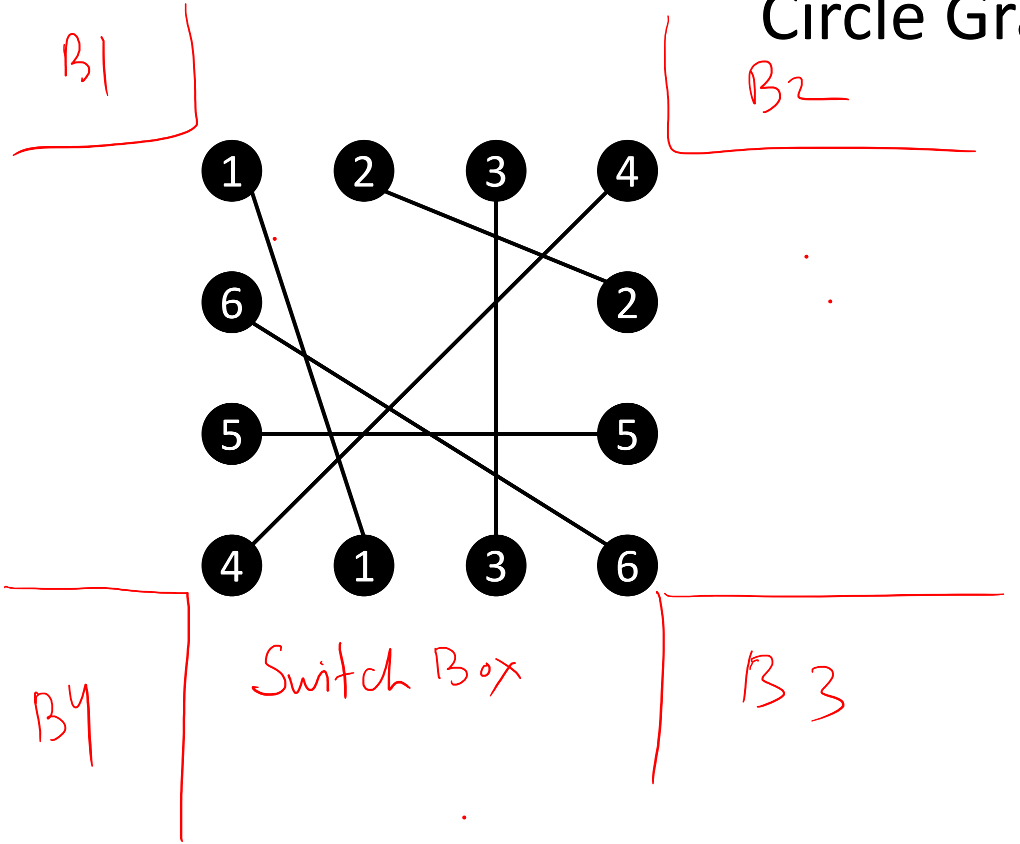
- **Matching diagram**: A matching diagram is a diagram where all the lines begin at a designated y-coordinate and end at another specified y-coordinate. This case typically occurs in channel routing.
- **Permutation graph** $G_P = (V, E_P)$, is defined as
  - $V = \{v_i \mid v_i \text{ represents interval } I_i\}$
  - $E_P = \{(v_i, v_j) \mid \text{if line } i \text{ intersects line } j\}$

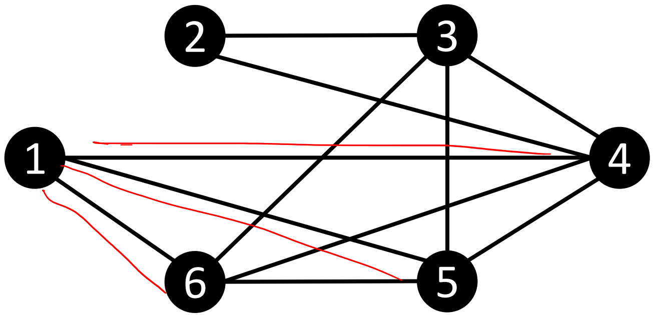
Permutation Graph





Circle Graph





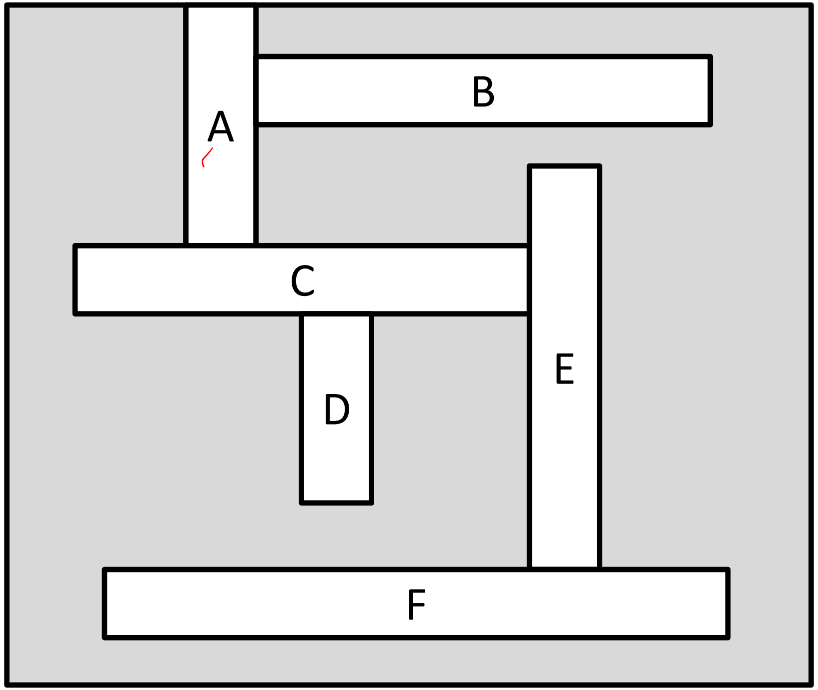
Graphs Related to a Set of Rectangles

- Given a set of rectangles $R = \{R_1, R_2, .., R_m\}$ corresponding to a layout in a plane, a **neighborhood graph** is a graph $G = (V, E)$, where
  - $V = \{v_i \mid v_i \text{ represents rectangle } R_i\}$
  - $E_C = \{(v_i, v_j) \mid R_i \text{ and } R_j \text{ are neighbors}\}$

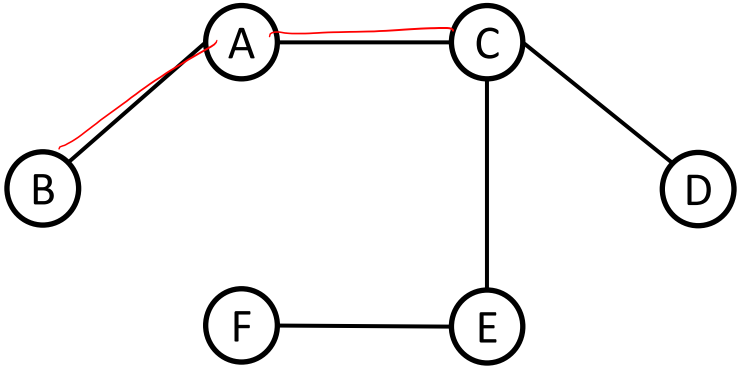
Graphs Related to a Set of Rectangles

- The neighborhood graph is used in the global routing phase of design automation.
  - Each channel is depicted as a rectangle
  - Channels are considered neighbors if they share a boundary.

Neighborhood Graph



VLSI Physical design



Graph Problems Related to Physical Design

- **Independent Set Problem:**
  - Instance: Graph $G = (V, E)$, positive integer $K \leq |V|$.
  - Question: Does G contain an independent set of size K or more, i.e., a subset $V' \subset V$ such that $|V'| \geq K$ and such that no two vertices in V' are joined by an edge in E?
- The problem is NP-complete for general graphs.
- The problem is solvable in polynomial time for interval, permutation, and circle graphs.

Graph Problems Related to Physical Design

- **Graph K – Colorability:**
  - Instance: Graph $G = (V, E)$, positive integer $K \leq |V|$.
  - Question: Is G K- colorable, i.e., does there exist a function $f : V \rightarrow \{1, 2, .., K\}$ such that $f(u) \neq f(v)$ whenever $\{u, v\} \in E$?
- The problem is NP-complete for general graphs and remains so for all fixed $K \geq 3$.
- It is polynomial for $K = 2$.

Graph Problems Related to Physical Design

- **Clique Problem:**
  - Instance: Graph $G = (V, E)$, positive integer $K \leq |V|$.
  - Does G contain a clique of size K or more, i.e., a subset $V' \subset V$ such that $|V'| \geq K$ and such that every two vertices in V' are joined by an edge in E ?
- The problem is NP-complete for general graphs.
- the problem is solvable in polynomial time for
  - chordal, interval, and comparability graphs

Thank You