Resource Allocation and Binding

Dr. Chandan Karfa

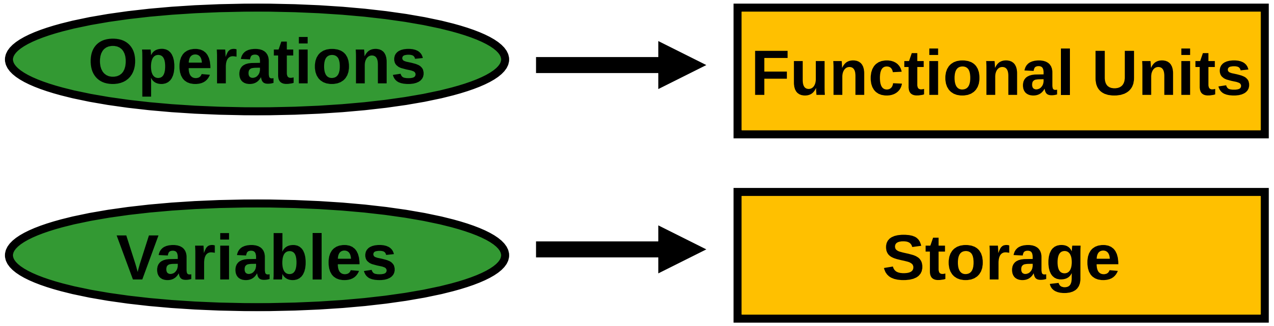
Department of Computer Science and Engineering

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

Recap: Allocation and Binding

- Objectives: Maximize Resource sharing; hence, minimize resource usage



Subtasks:

1. FU allocation & Binding
2. Register Allocation & Binding

Resource Optimization as Clique Cover Problem

- A group of mutually compatible operations corresponds to a subset of vertices that are all mutually connected by edges, i.e., to a clique.
- Therefore a ***maximal*** set of mutually compatible operations is represented by a ***maximal clique*** in the compatibility graph.
- An optimum resource sharing is one that minimizes the number of required resource instances.
- We can associate a resource instance to each clique, the problem is equivalent to partitioning the graph into a minimum number of cliques. ***clique cover number*** of G+(V, E), denoted by *k*(G+(V, E)).

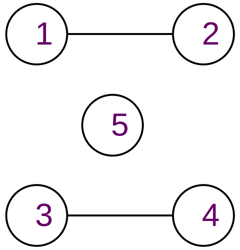
Resource Sharing as Graph Coloring Problem

- A set of mutually compatible operations corresponds to a subset of vertices that are not connected by edges, also called the ***independent set*** $(\alpha(G^+))$
- A proper vertex coloring of the conflict graph provides a solution to the sharing problem
- Each color corresponds to a resource instance
- An optimum resource sharing corresponds to a vertex coloring with a minimum number of colors. Such a number is the chromatic number of G-(V, ***E)***

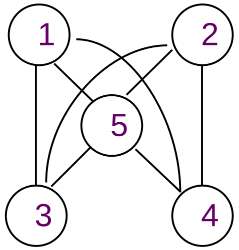
Example

t1	x=a+b y=c+d	1 2
t2	s=x+y t=x-y	3 4
t3	z=a+t	5

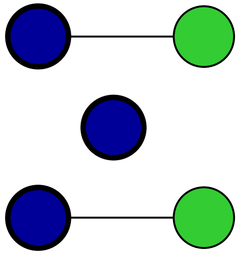
Conflict



Compatibility



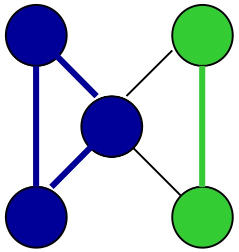
Coloring



ALU1: 1,3,5

ALU2: 2,4

Partitioning

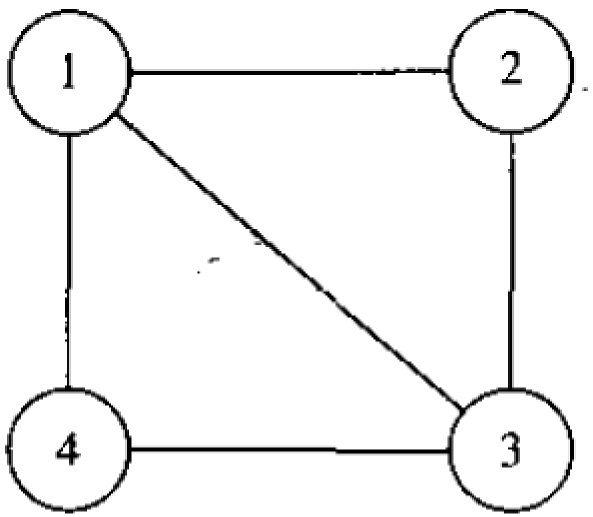


Compatibility and conflicts

- Compatibility graph:
  - Partition the graph into a minimum number of cliques
  - Find clique cover number κ (G_+) → No of cliques
- Conflict graph:
  - Color the vertices by a minimum number of colors.
  - Find the chromatic number χ (G_-)
- NP-complete problems
  - Heuristic algorithms

Chordal Graph (A special set of Perfect Graph)

- A graph is said to be ***chordal,*** or ***triangulated,*** if every cycle with more than three edges possesses a ***chord,*** i.e., an edge joining two non-consecutive vertices in the cycle

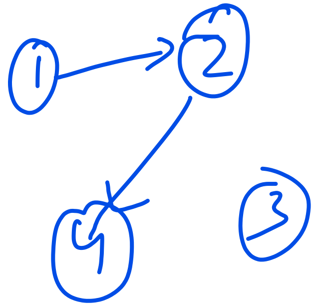


Interval Graph

- A subclass of chordal graphs is the one of ***interval graphs.***
- An interval graph is a graph whose vertices can be put in one-to-one correspondence with a set of ***intervals,*** so that two vertices are adjacent if and only if the corresponding intervals intersect.
- Advantage: The graph coloring problem is polynomial time solvable
- How to identify a graph is an interval graph?

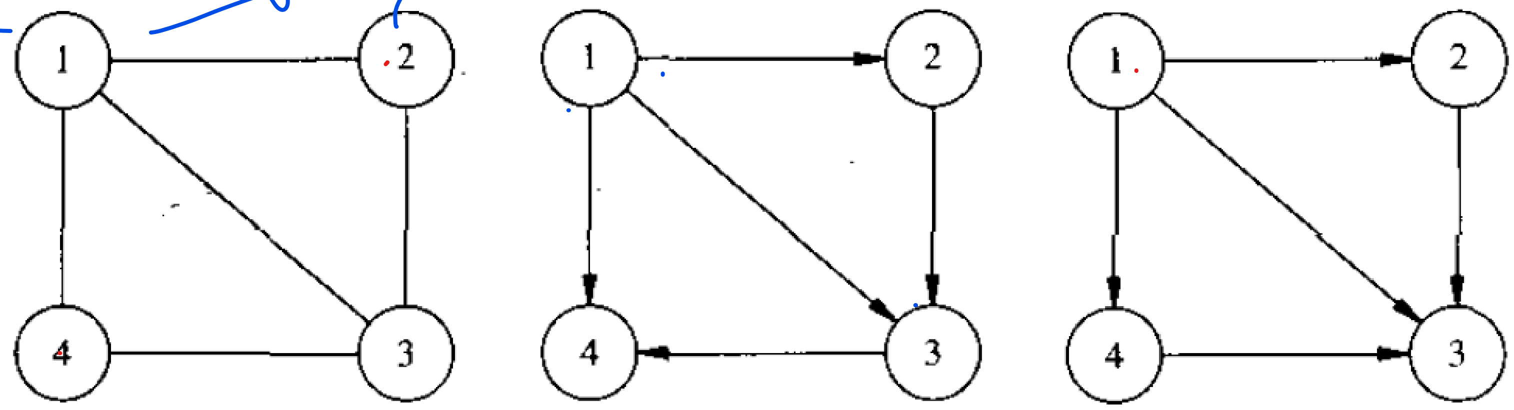
Comparability graph

- A graph ***G(V, F)*** is a ***comparability graph*** if it satisfies the ***transitive orientation property,*** i.e., if it has an orientation such that in the resulting directed graph **G(V, E*),*** **((vi, vj) in E** and **(vj, vk) in E implies (vi, vk) in E.**



Con

conflict



(a) (b) (c)

Gilmore Theorem

- An important theorem, by Gilmore and Hoffman, relates comparability to interval graphs

- **Theorem:** An undirected graph is an interval graph if and only if it is chordal and its complement is a comparability graph

→ Tran

Data-flow graphs
(flat sequencing graphs)

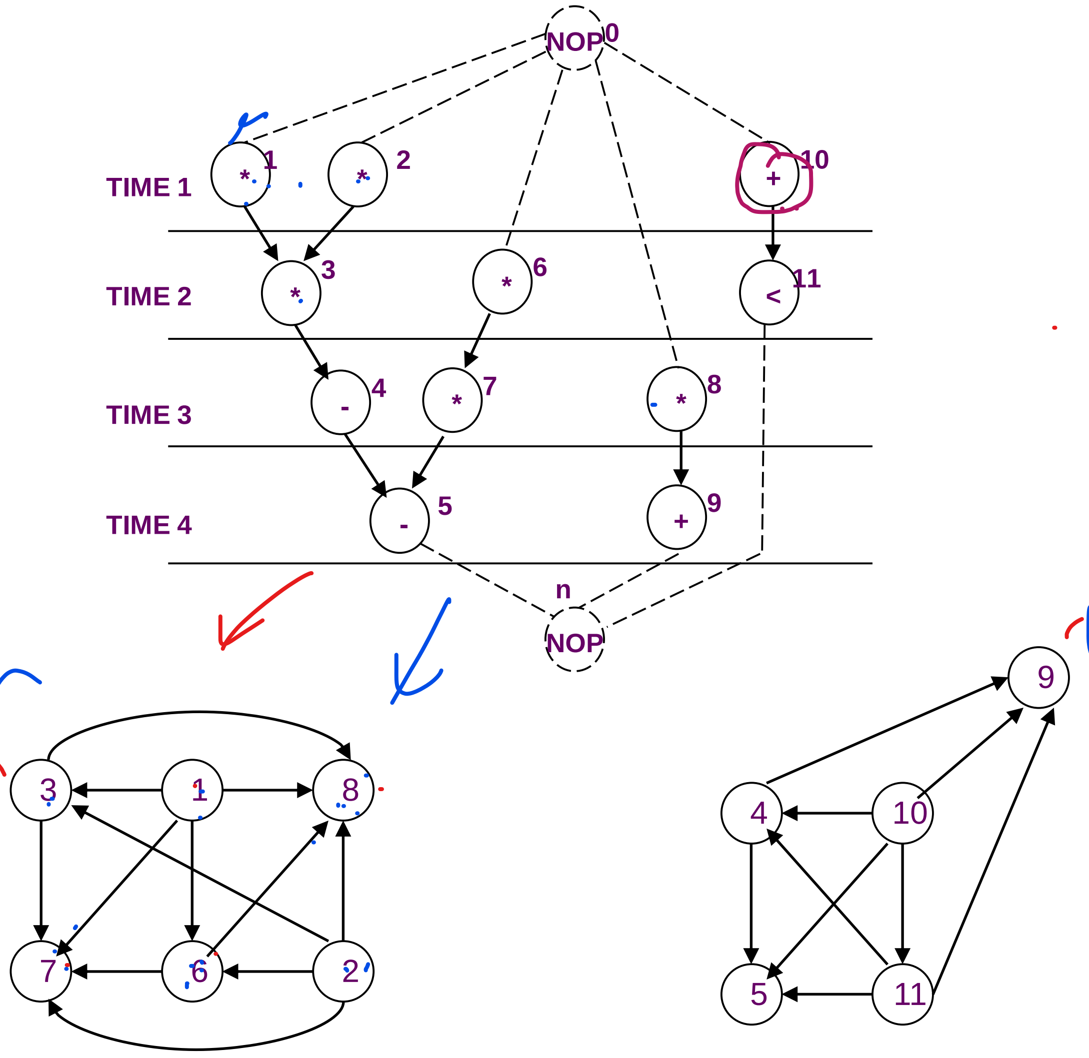
- The compatibility/conflict graphs have special properties:
  - Compatibility
    - Comparability graph
  - Conflict
    - Chordal
  - So they are perfect graph satisfying Gilmore's theorem
- Polynomial time solutions for Interval Graph
  - Left-edge algorithm

Imp

Example

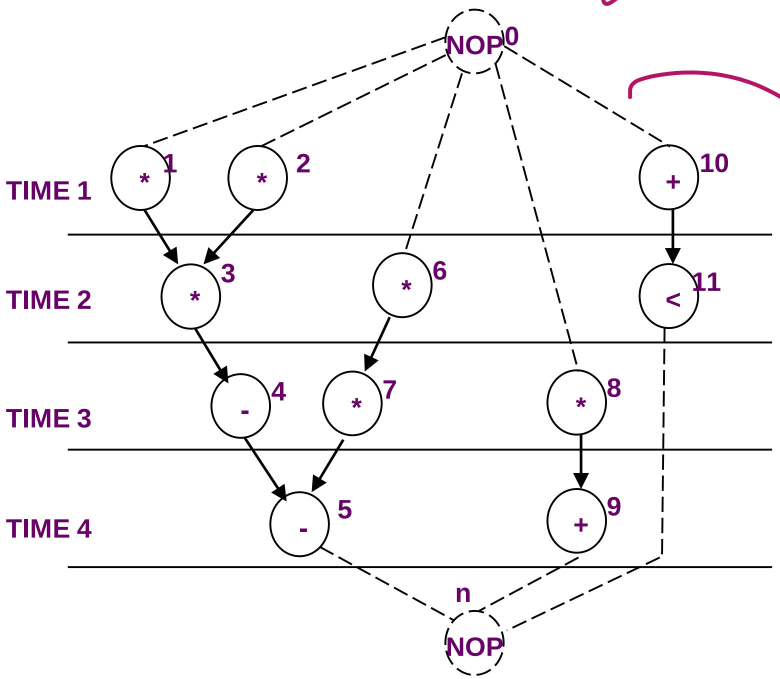
Comparability graph

Mul

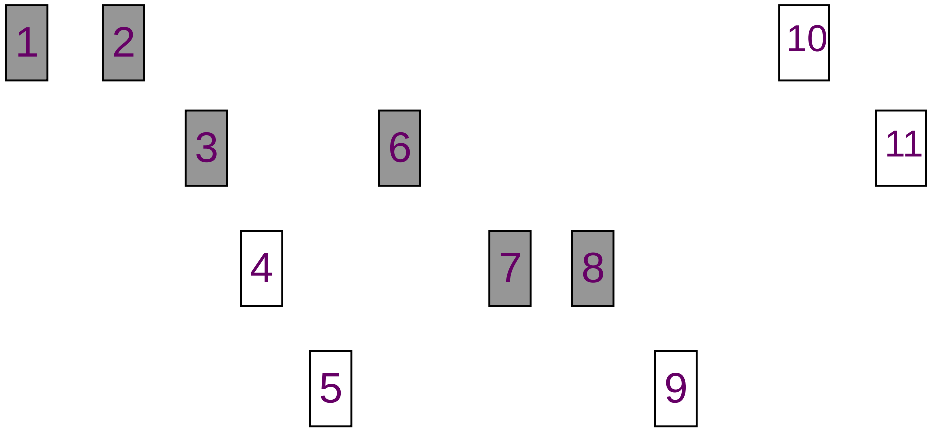


ALU

Example



NOP 0
TIME 1
TIME 2
TIME 3
TIME 4
* 1
* 2
+ 10
* 3
* 6
< 11
- 4
* 7
* 8
- 5
+ 9
n
NOP



1
2
10
3
6
11
4
7
8
5
9

Left-edge algorithm

- **Input:**
  - Set of intervals with *left* and *right edge*
    - *Start and Stop times*
  - A set of *colors* (initially one color)
- **Rationale:**
  - Sort intervals in a *list* by *left* edge
  - Assign non overlapping intervals to first color using the list
  - When possible intervals are exhausted,
    increase color counter and repeat

Example

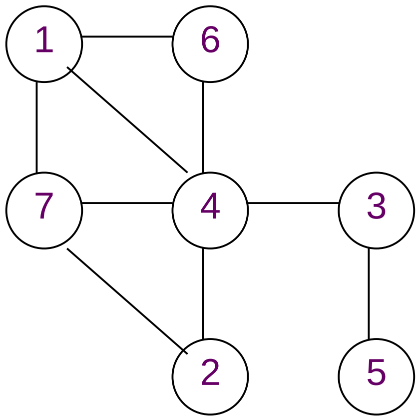
No of cliques
↳ ch. no compatible

$\alpha(G) \geq G^+$
G^-

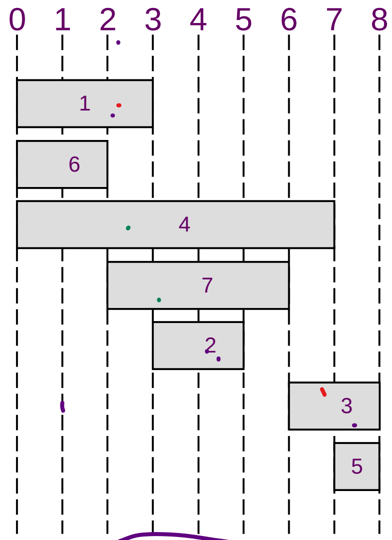
MIS

$\{1, 2, \ldots\}$

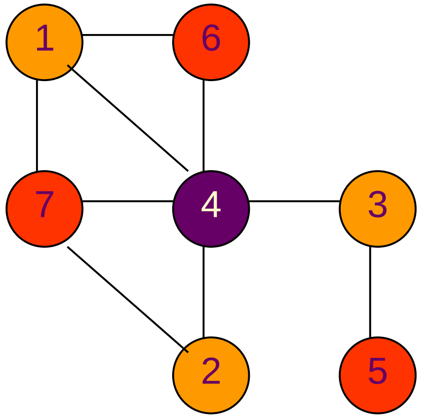
3



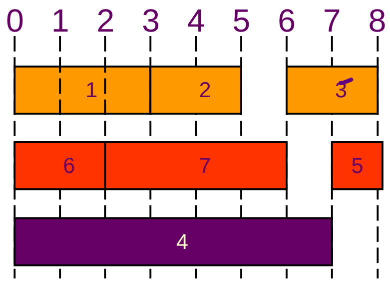
Conflict graph



Intervals



Colored conflict graph



Coloring

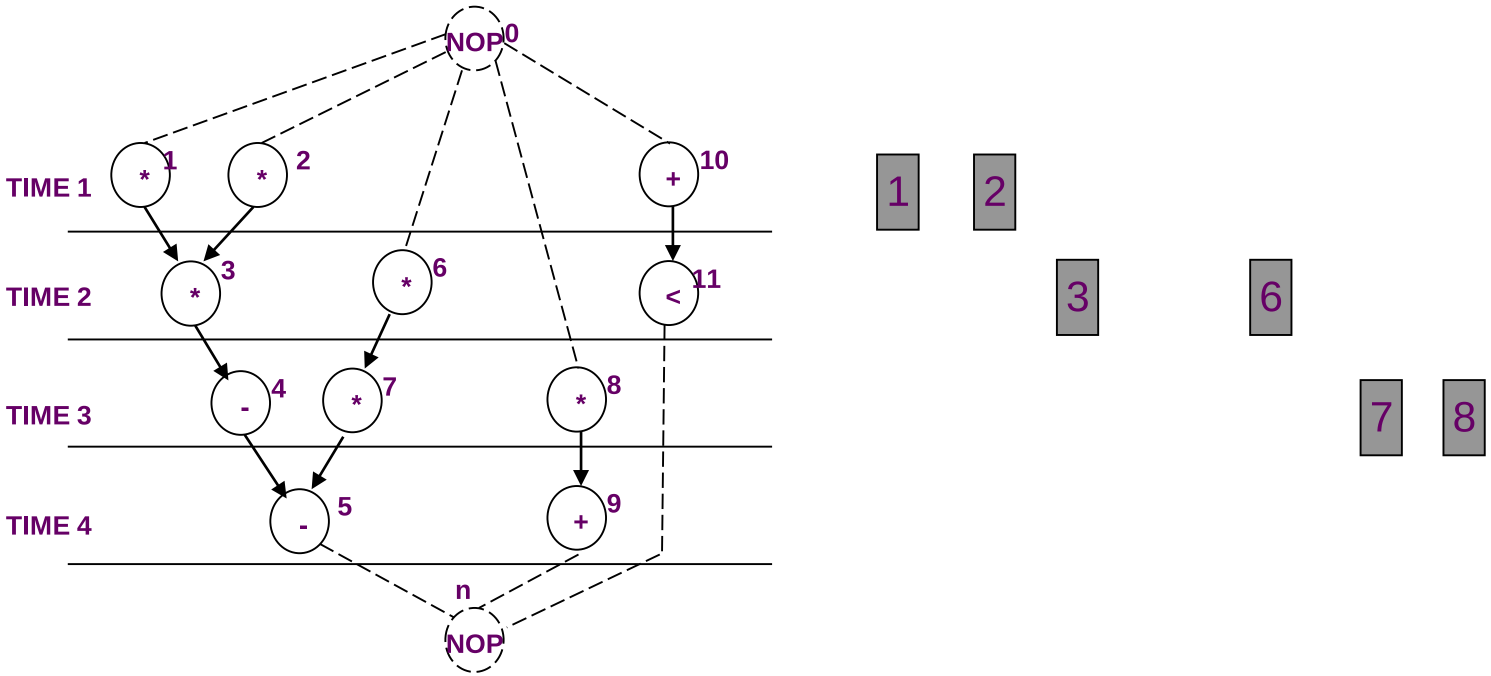
Left-edge algorithm

```
LEFT_EDGE(I) {
    Sort elements of I in a list L in ascending order of l_i;
    c = 0;
    while (some interval has not been colored) do {
        S = Ø;
        r = 0;
        while ( exists s є L such that l_s > r) do {
            s = First element in the list L with l_s > r;
            S = S U {s};
            r = r_s;
            Delete s from L;
        }
        c = c + 1;
        Label elements of S with color c;
    }
}
```

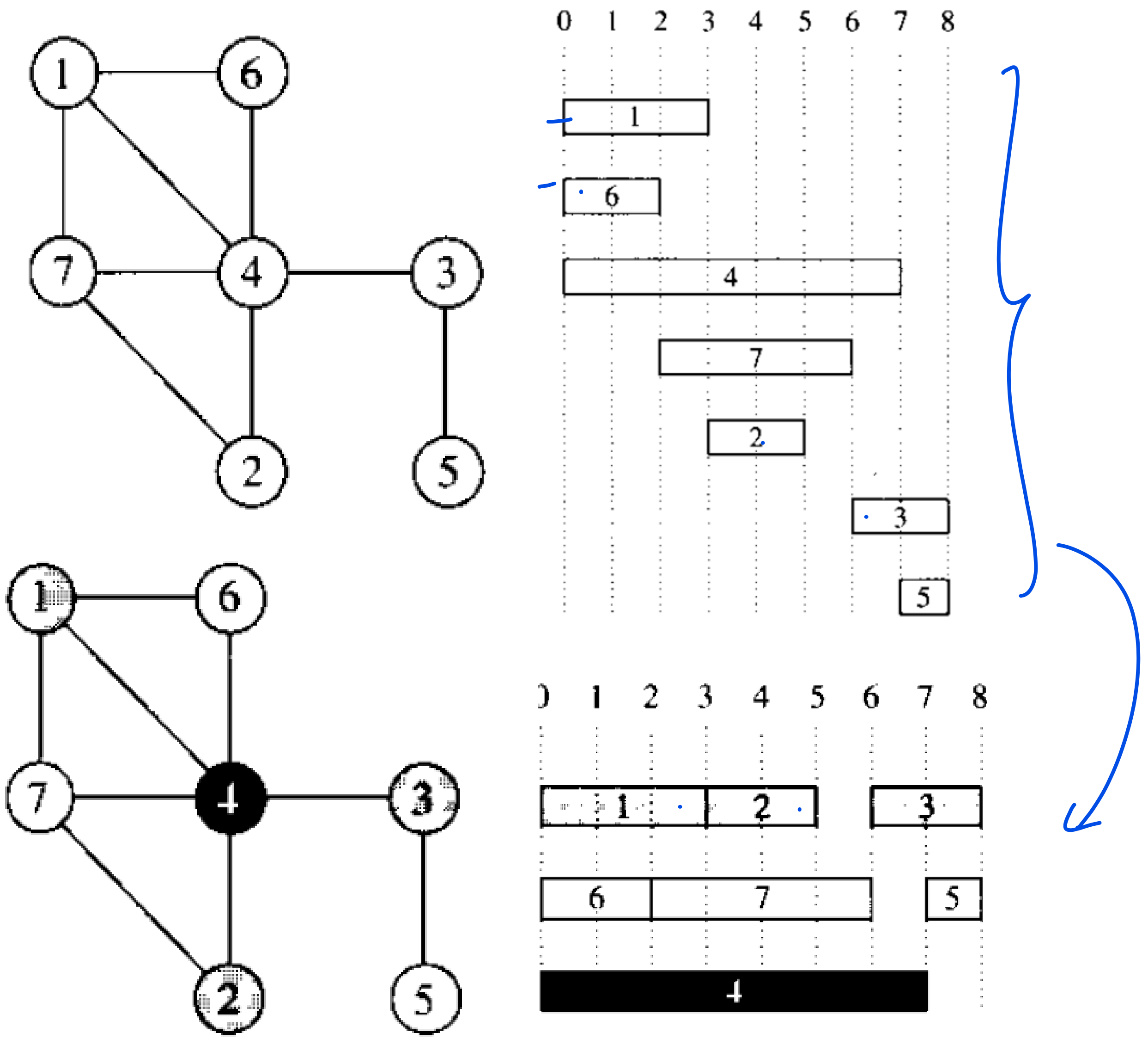
Complexity

- O(|V| log (|V|), |V| is the number of nodes
  - Sorting

Example



An Example



Thank You