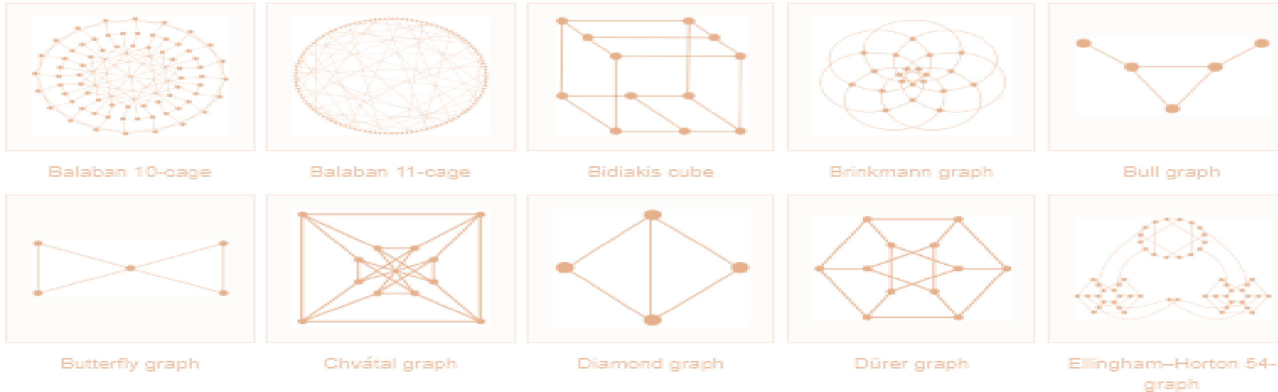
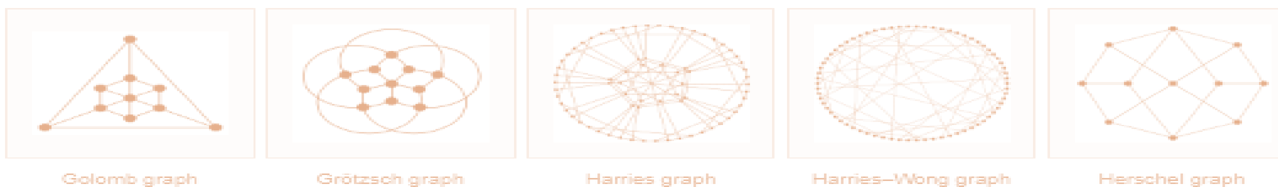
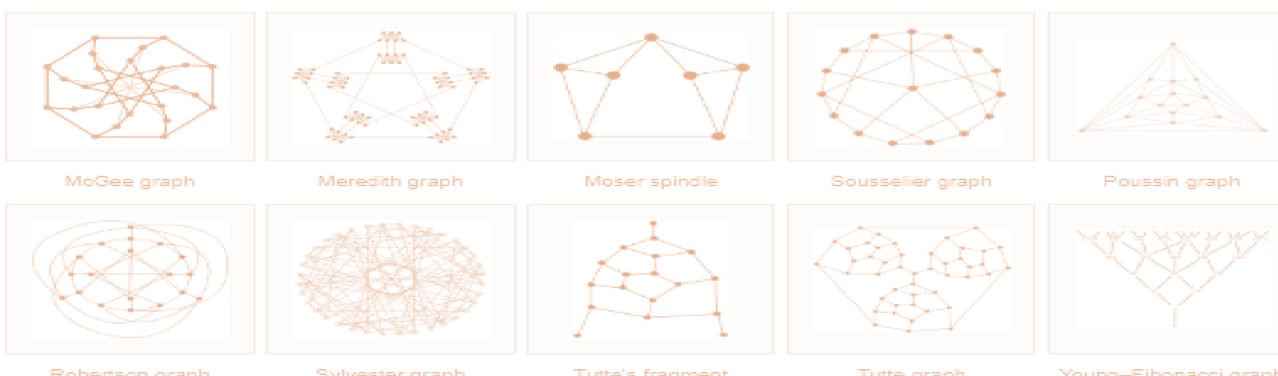
Graph Theory



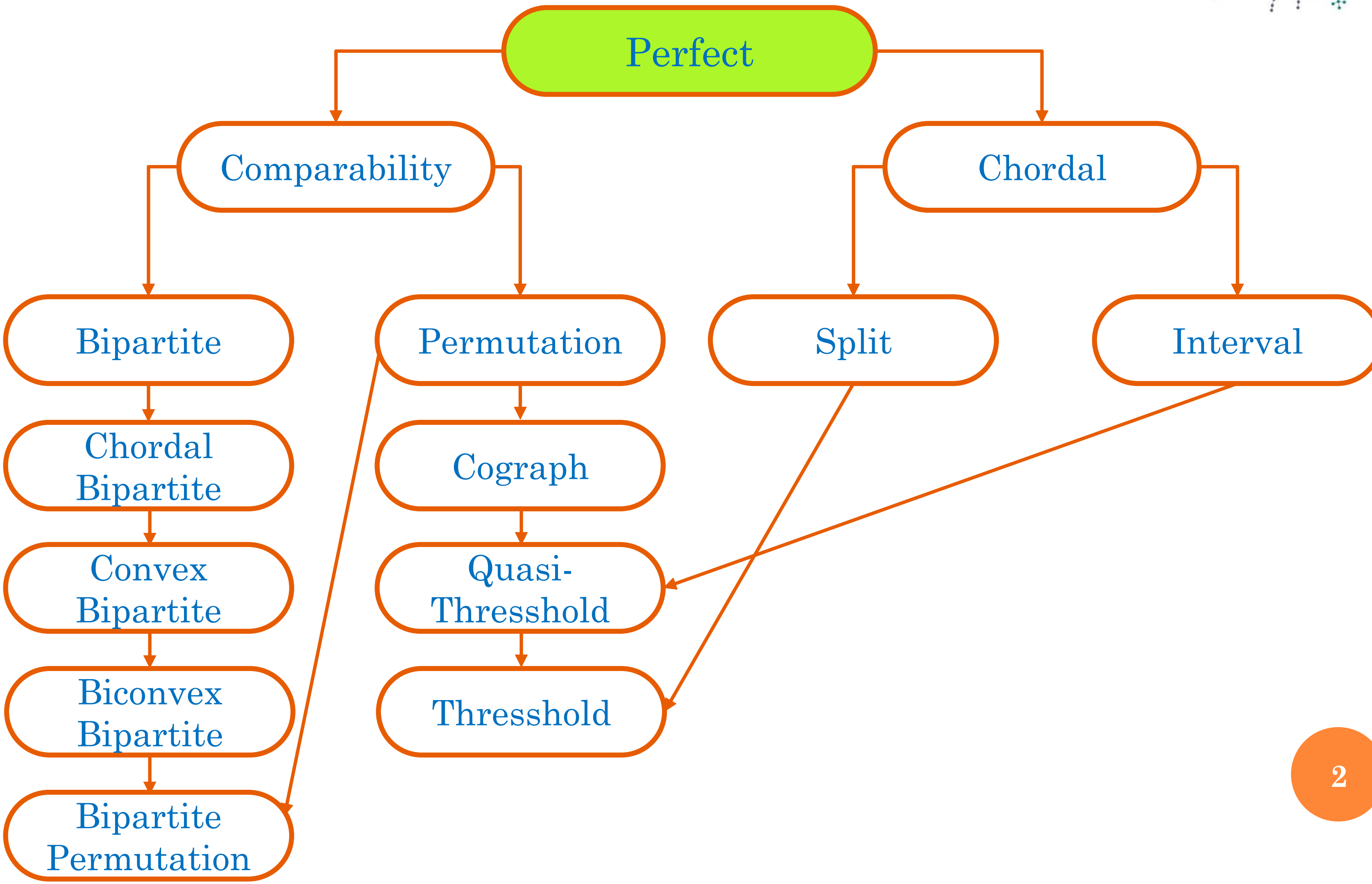


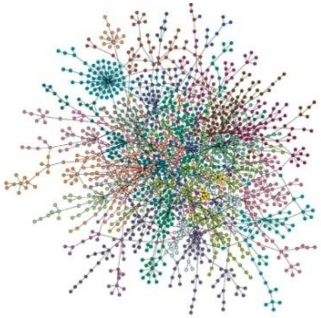
Introduction to Graph Theory (B)



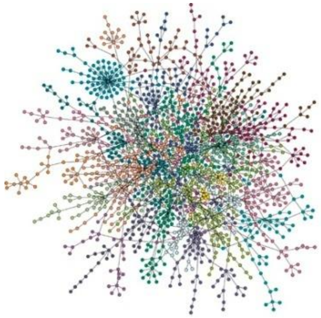
BASICS…

- **Graph Classes**

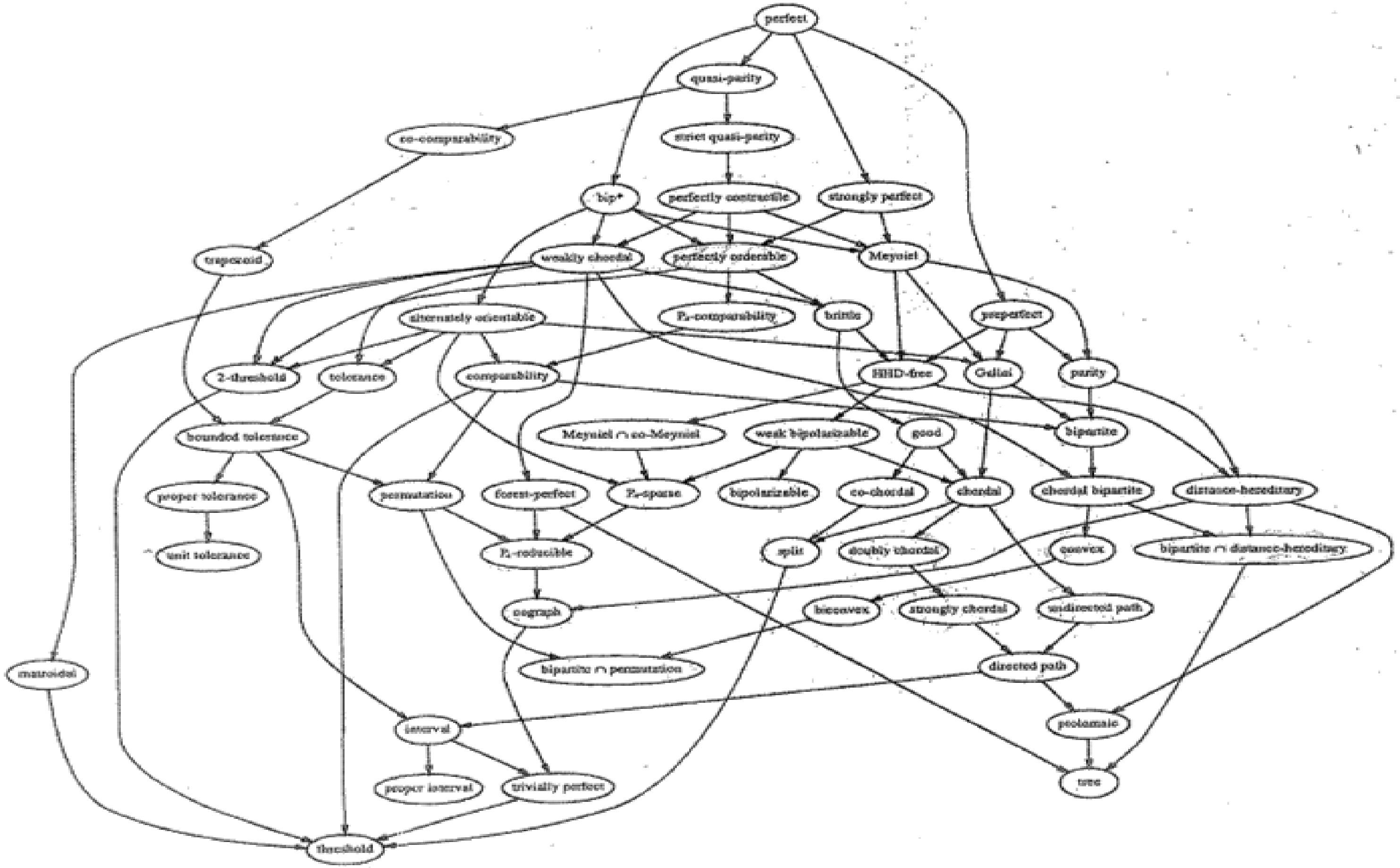




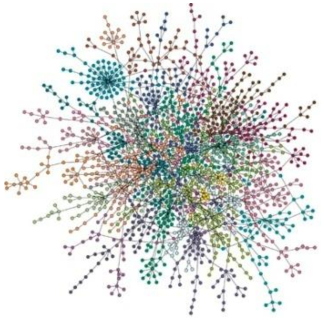
Basics…



- **Graph Classes**



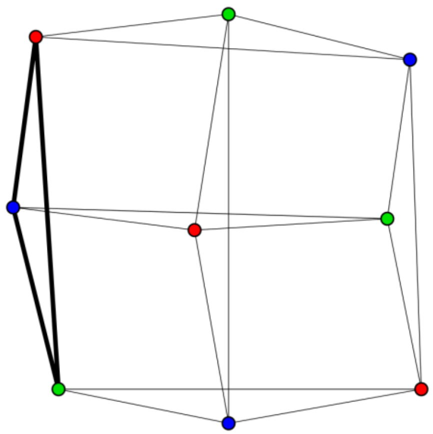
BASICS…



Graph Class: Perfect Graphs

- A **Perfect Graph** is a graph in which the chromatic number of every induced subgraph equals the size of the largest clique of that subgraph .
- An arbitrary graph G is perfect if and only if we have:

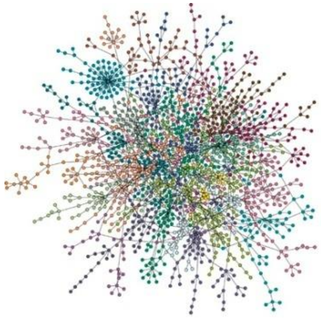
$$\forall S \subseteq V(G)(\chi(G[S]) = \omega\,(G[S]))$$

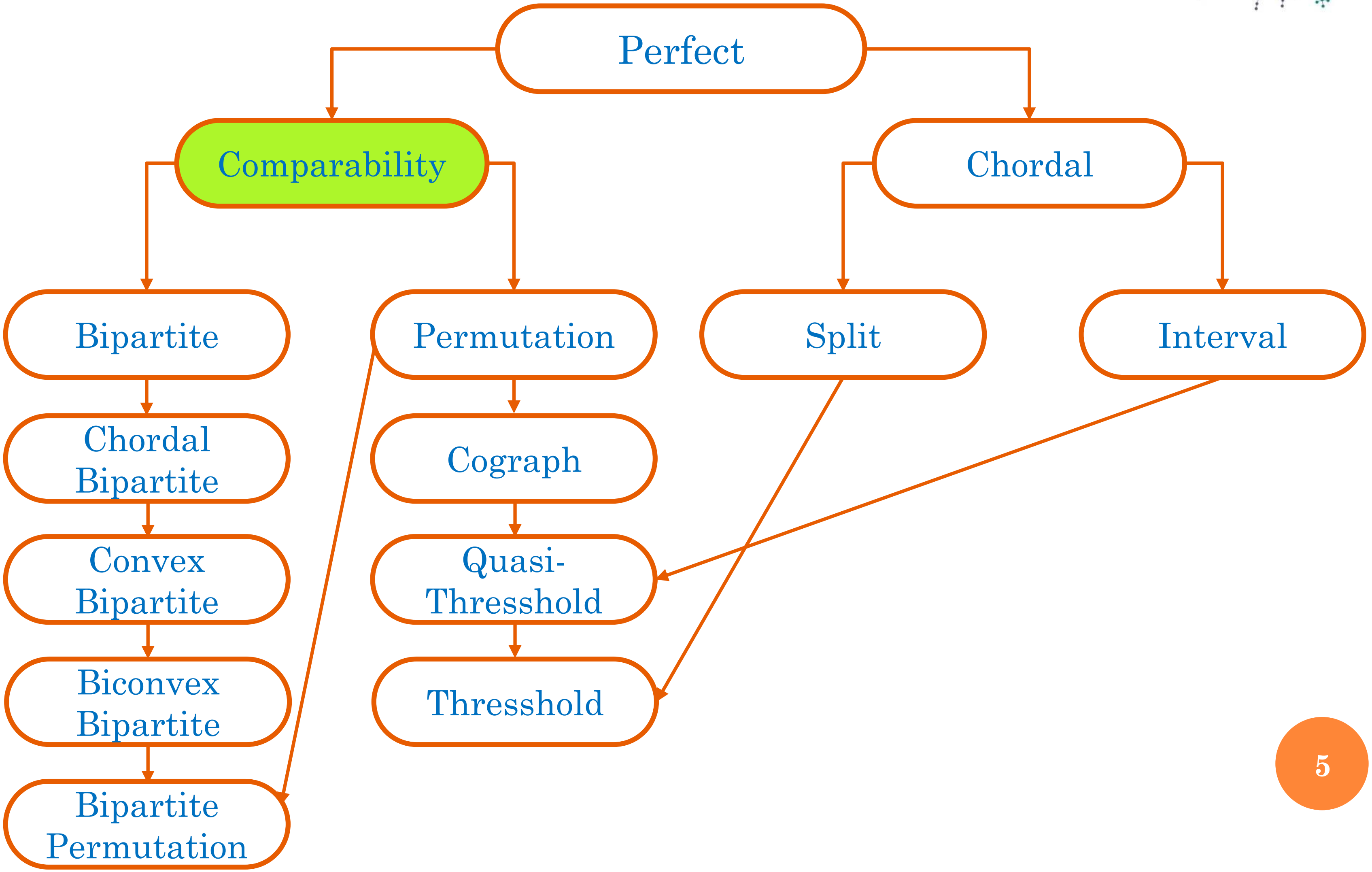


- A graph G is perfect ***iff*** its complement $\bar{G}$ is perfect (perfect graph theorem)
- A perfect graphs are the same as Berge graphs, which are graphs G where neither G nor $\bar{G}$ contain an induced cycle of odd length 5 or more (strong perfect graph theorem).

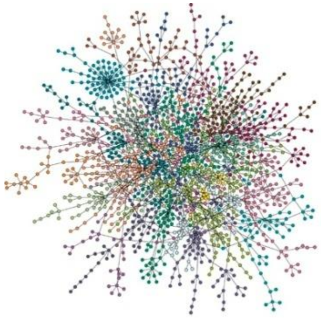
BASICS…

- **Graph Classes**



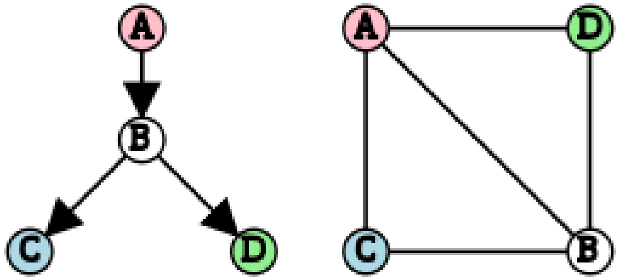


Basics…

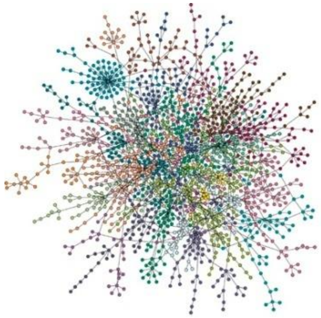


- **Graph Class: Comparability Graphs**
  - A **comparability graph** is an undirected graph that connects pairs of elements that are comparable to each other in a partial order.
  - Comparability graphs have also been called **transitively orientable graphs**, partially orderable graphs, containment graphs, and divisor graphs.
  - An incomparability graph is an undirected graph that connects pairs of elements that are not comparable to each other in a partial order.
  - Satisfy the **Transitive Orientation Property**

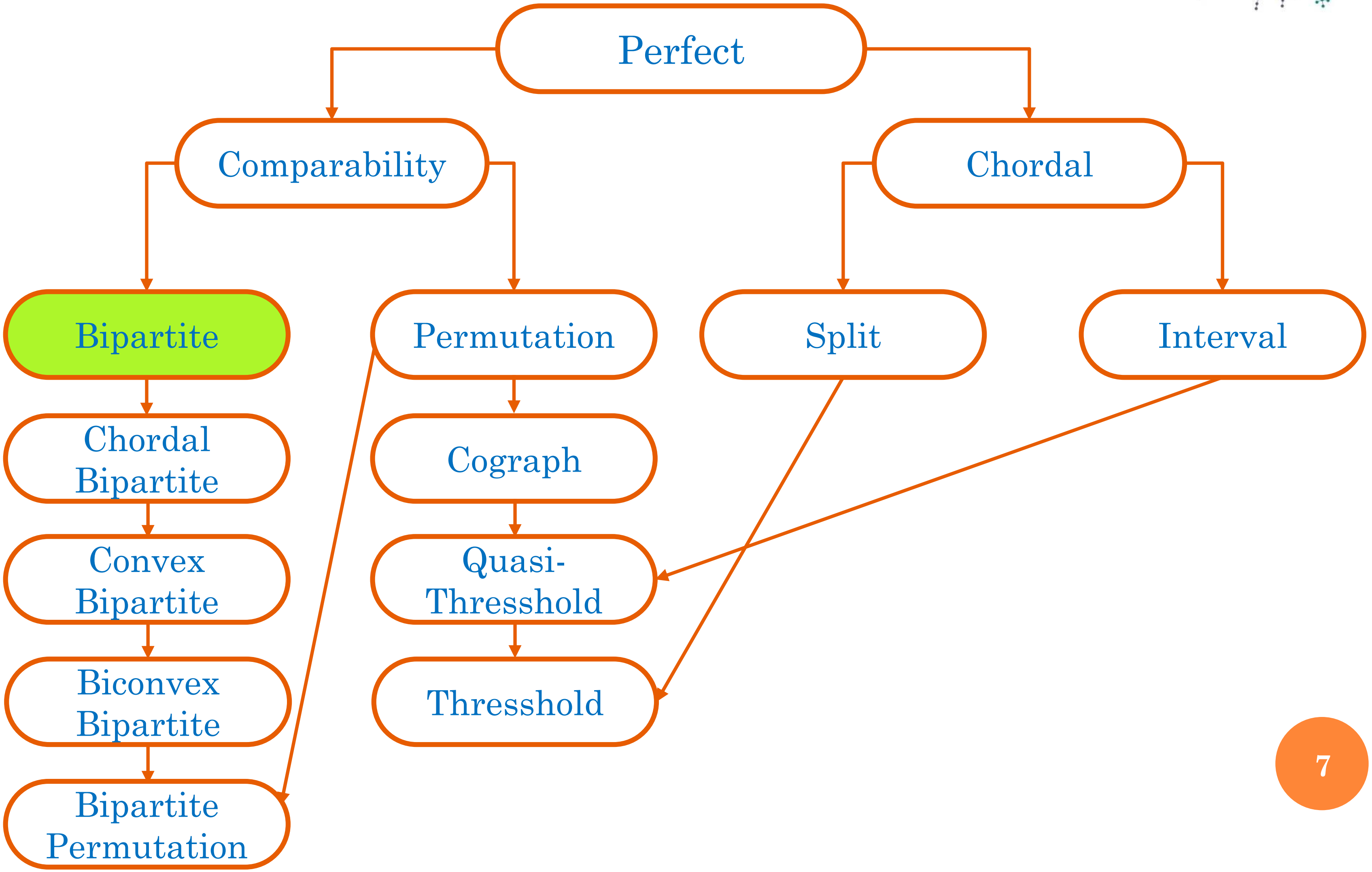
    Each edge can be assigned a one-way direction in such a way that the resulting oriented graph (V, F): $ab \in F \text{ and } bc \in F \rightarrow ac \in F \ (\forall a, b, c \in V)$



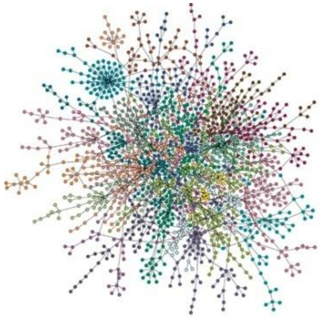
Basics…



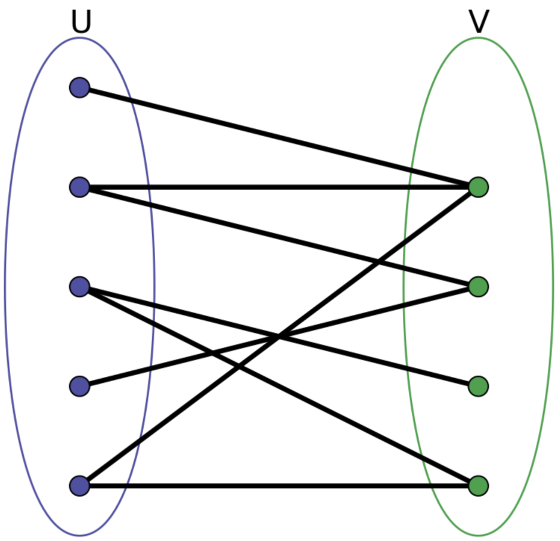
- **Graph Classes**



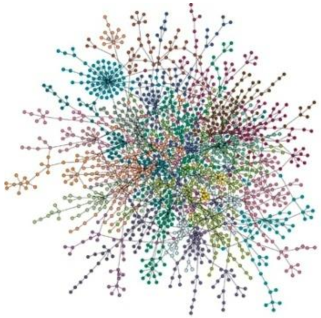
Basics…



- **Graph Class: Bipartitite Graphs**
  - In a simple graph G, if V can be partitioned into two disjoint sets V_1 and V_2 such that every edge in the graph connects a vertex in V_1 and a vertex V_2
  - Remark: no edge in G connects either two vertices in V_1 or two vertices in V_2

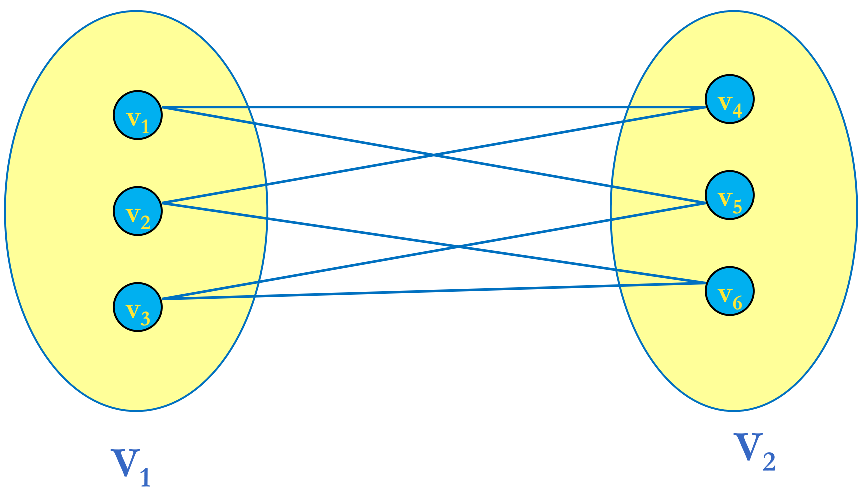


Basics…

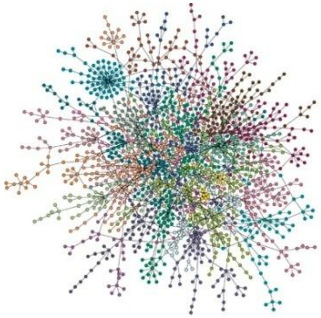


Graph Class: Bipartitite Graphs

- In a simple graph G, if V can be partitioned into two disjoint sets V_1 and V_2 such that every edge in the graph connects a vertex in V_1 and a vertex V_2
- Remark: no edge in G connects either two vertices in V_1 or two vertices in V_2
- Application example: Representing Relations
- Representation example: $V_1 = \{v_1, v_2, v_3\}$ and $V_2 = \{v_4, v_5, v_6\}$,

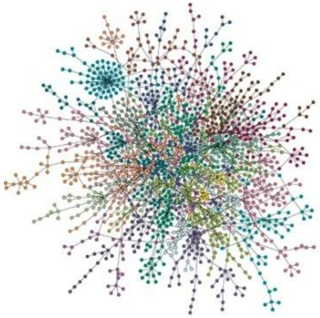


Basics…

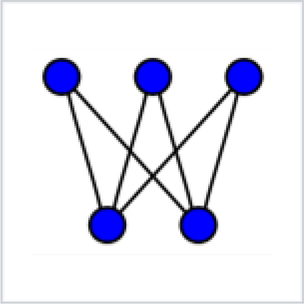


- **Graph Class: Bipartitite Graphs (Complete)**
  - A **complete bipartite graph** has its vertex set portioned into two subsets of m and n vertices, respectively.
  - There is an edge between two vertices if and only if one vertex is in the first subset and the other vertex is in the second subset.
  - The complete bipartite graph is usually denoted $K_{n,m}$

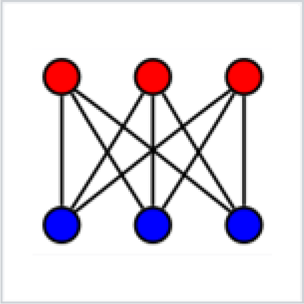
BASICS…



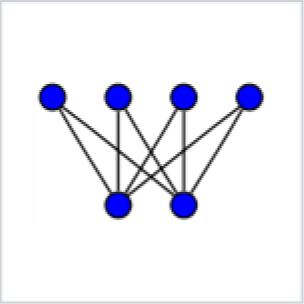
- **Graph Class: Bipartitite Graphs (Complete)**
  - A **complete bipartite graph** has its vertex set portioned into two subsets of m and n vertices, respectively.
  - There is an edge between two vertices if and only if one vertex is in the first subset and the other vertex is in the second subset.
  - The complete bipartite graph is usually denoted $K_{n,m}$



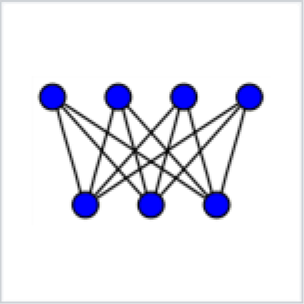
$K_{2,3}$



$K_{3,3}$.

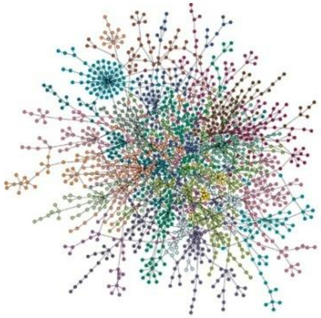


$K_{2,4}$



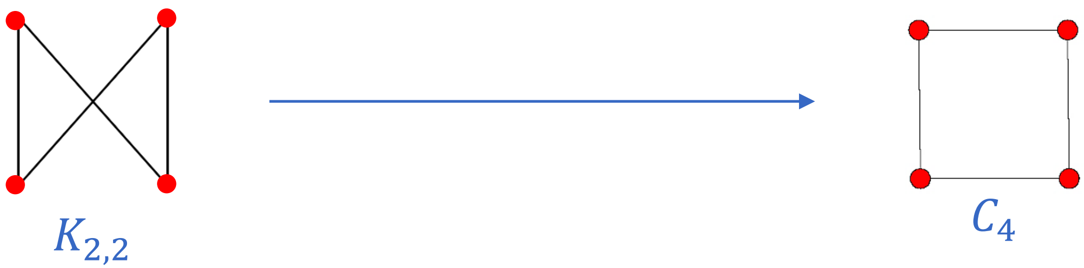
$K_{3,4}$

Basics…



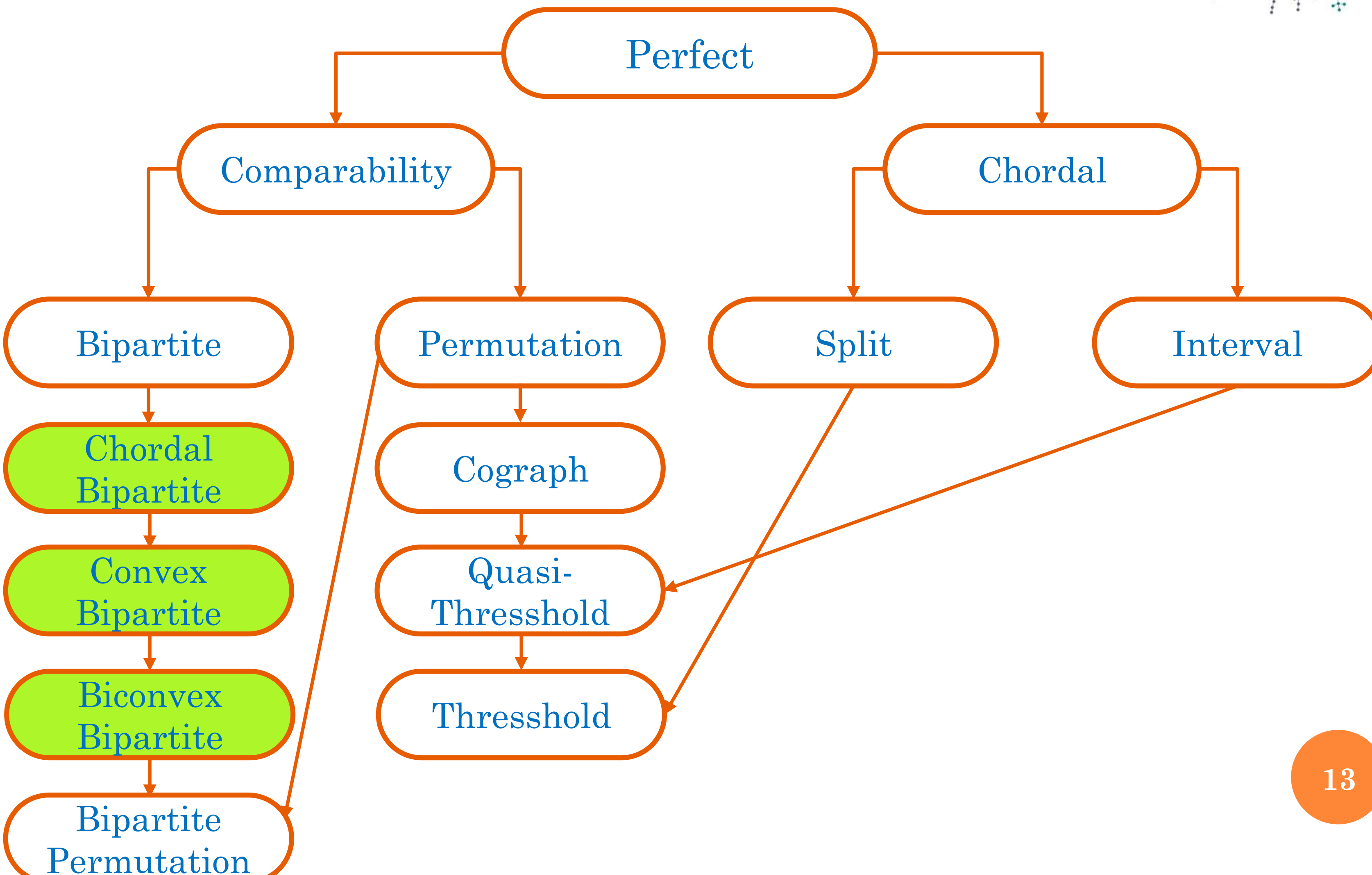
Graph Class: Bipartitite Graphs (Complete)

- A **complete bipartite graph** has its vertex set portioned into two subsets of m and n vertices, respectively.
- There is an edge between two vertices if and only if one vertex is in the first subset and the other vertex is in the second subset.
- The complete bipartite graph is usually denoted $K_{n,m}$
- The graph $K_{2,2}$ equals the 4-cycle C_4 (the square).

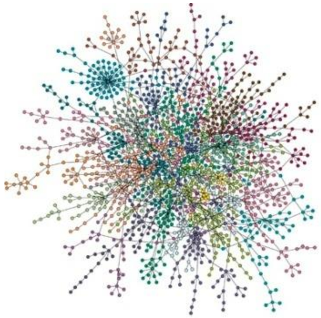


BASICS…

- **Graph Classes**



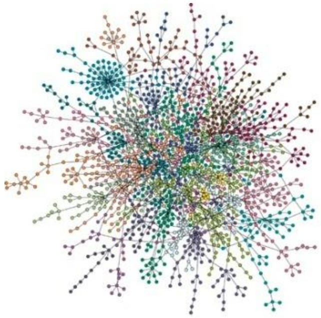
Basics…

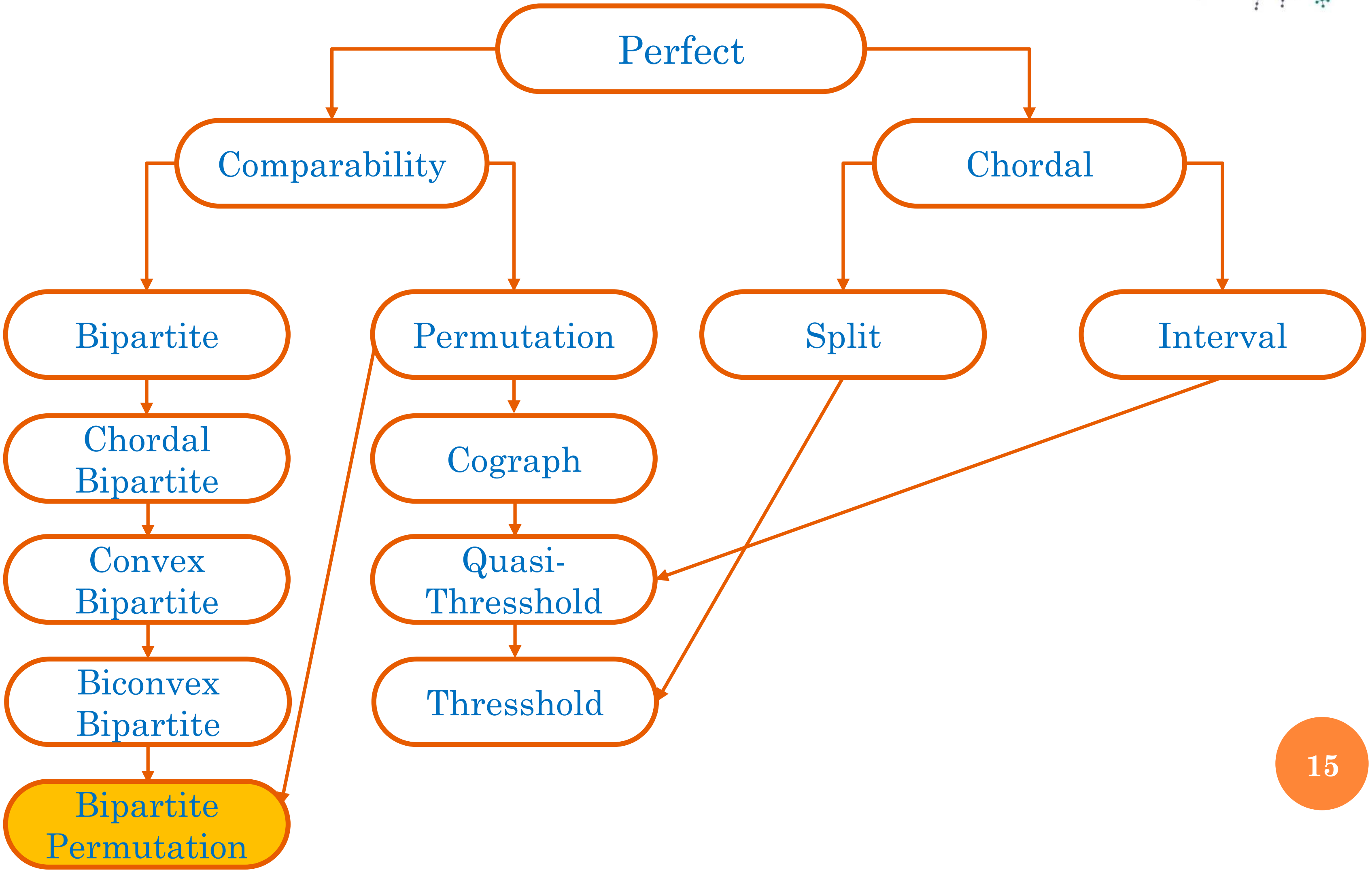


- **Graph Class: Sub-classes of Bipartitite Graphs**
  - In a simple graph G, if V can be partitioned into two disjoint sets V_1 and V_2 such that every edge in the graph connects a vertex in V_1 and a vertex V_2
  - Remark: no edge in G connects either two vertices in V_1 or two vertices in V_2
  - A **chordal bipartite graph** is a bipartite graph $B = (X, Y, E)$ in which every cycle of length at least 6 in B has a chord, i.e., an edge that connects two vertices that are a distance > 1 apart from each other in the cycle
  - A **convex bipartite graph** is a bipartite graph with specific properties. A bipartite graph, $(U \cup V, E)$, is said to be convex over the vertex set U if U can be enumerated such that for all $v \in V$ the vertices adjacent to v are consecutive.
  - Convexity over V is defined analogously. A bipartite graph (U ∪ V, E) that is convex over both U and V is said to be **biconvex** or doubly convex.

Basics…

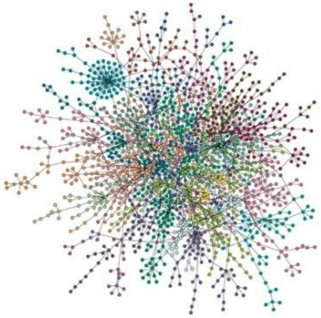
- **Graph Classes**

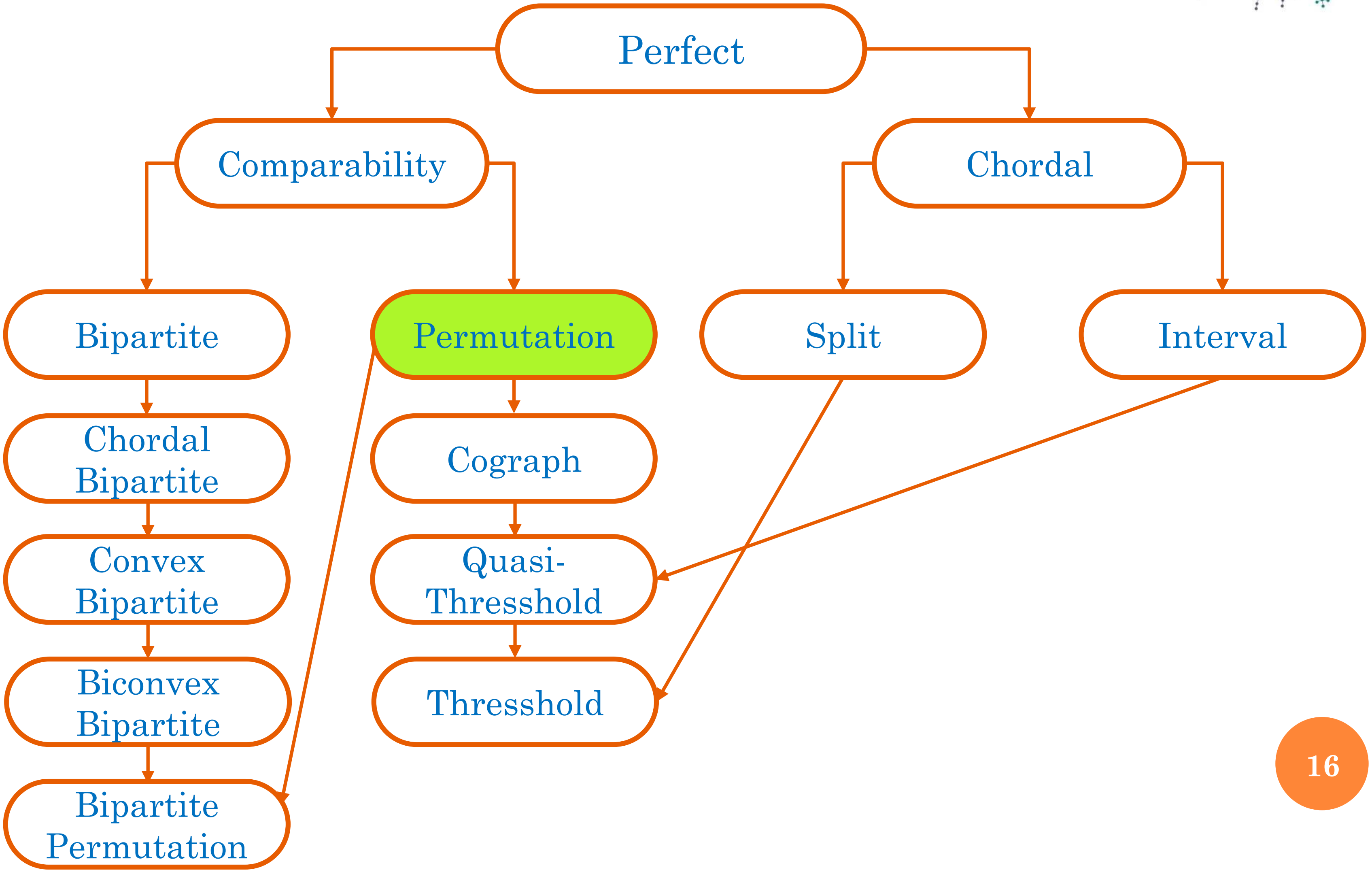




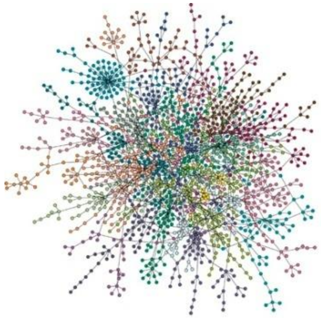
Basics…

- **Graph Classes**





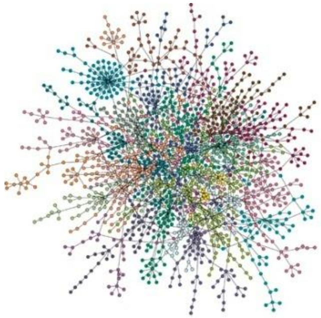
BASICS…



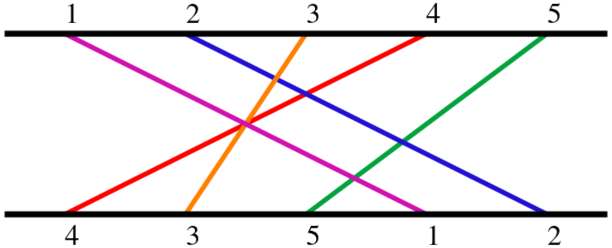
Graph Class: Permutation Graphs

- An **intersection graph** is a graph that represents the pattern of intersections of a family of sets. Any graph can be represented as an intersection graph
- A **permutation graph** is a graph whose vertices represent the elements of a permutation, and whose edges represent pairs of elements that are reversed by the permutation.
  - Geometrically can be defined, as the intersection graphs of line segments whose endpoints lie on two parallel lines.

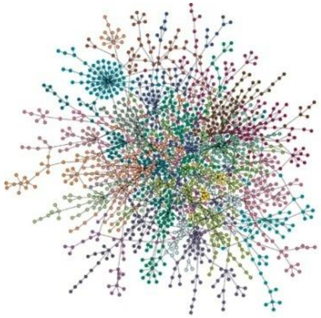
BASICS…



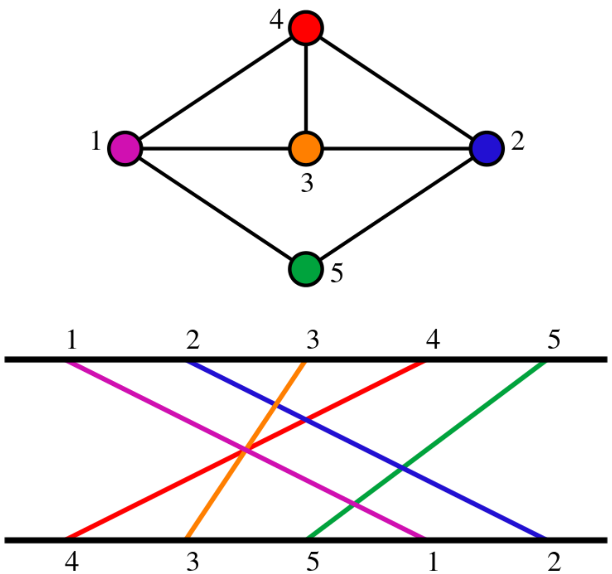
- **Graph Class: Permutation Graphs**
  - An **intersection graph** is a graph that represents the pattern of intersections of a family of sets. Any graph can be represented as an intersection graph
  - A **permutation graph** is a graph whose vertices represent the elements of a permutation, and whose edges represent pairs of elements that are reversed by the permutation.
    - Geometrically can be defined, as the intersection graphs of line segments whose endpoints lie on two parallel lines.



Basics…

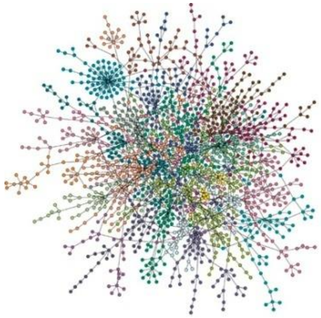


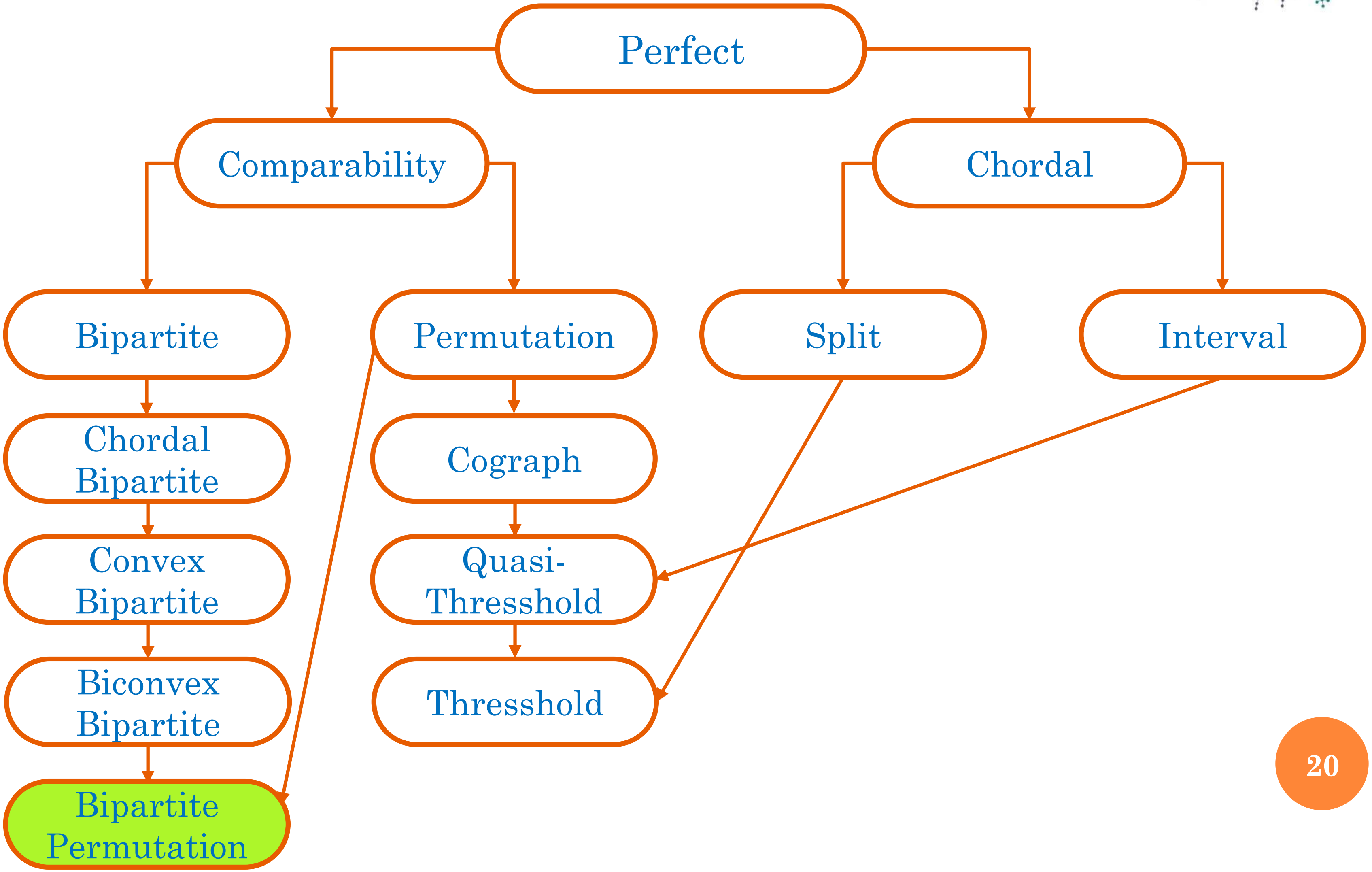
- **Graph Class: Permutation Graphs**
  - An **intersection graph** is a graph that represents the pattern of intersections of a family of sets. Any graph can be represented as an intersection graph
  - A **permutation graph** is a graph whose vertices represent the elements of a permutation, and whose edges represent pairs of elements that are reversed by the permutation.
    - Geometrically can be defined, as the intersection graphs of line segments whose endpoints lie on two parallel lines.



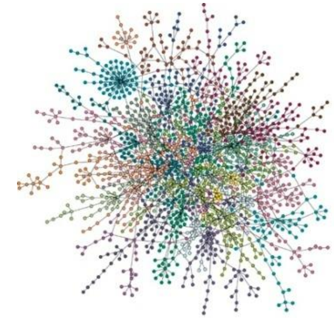
Basics…

- **Graph Classes**

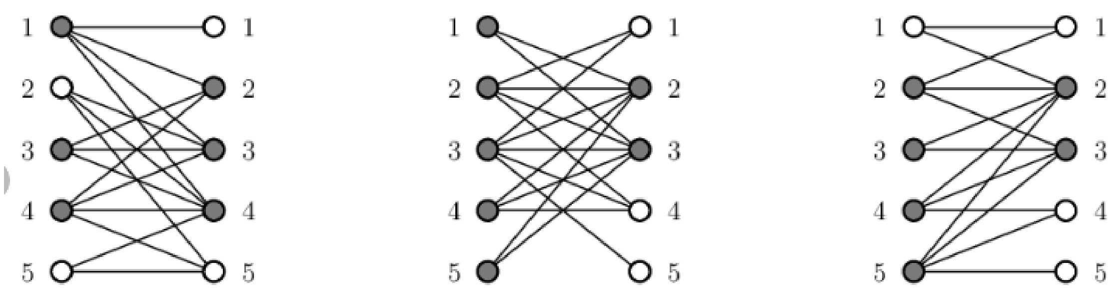




Basics…



- **Graph Class: Bipartite Permutation Graphs**
  - A graph is a **bipartite permutation graph**, if it is both bipartite and a permutation graph.
  - A bipartite graph is a bipartite permutation graph iff it admits a **strong ordering**.
  - A bipartite graph $G = (A, B, E)$ is a bipartite permutation graph iff it admits an ordering of A that has the **adjacency** and **enclosure properties**

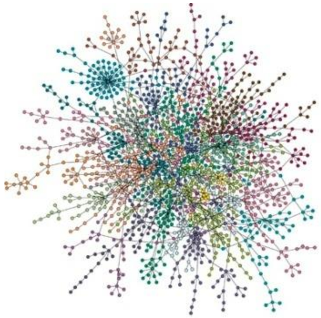


(a) Convex Bipartite

(b) BiConvex Bipartite

(c) Bipartite Permutation

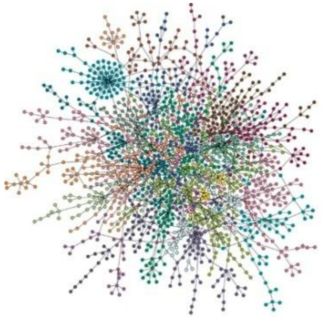
BASICS…



- **Graph Class: Bipartite Permutation Graphs**
  - A graph is a **bipartite permutation graph**, if it is both bipartite and a permutation graph.
  - A bipartite graph is a bipartite permutation graph iff it admits a **strong ordering**.
  - A bipartite graph $G = (A, B, E)$ is a bipartite permutation graph iff it admits an ordering of A that has the **adjacency** and **enclosure properties**

How many classes could be defined,
if we combine properties from different graph classes…

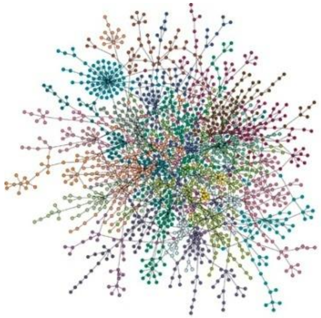
Basics…



- **Graph Class: Bipartite Permutation Graphs**
  - A graph is a **bipartite permutation graph**, if it is both bipartite and a permutation graph.
  - A bipartite graph is a bipartite permutation graph iff it admits a **strong ordering**.
  - A bipartite graph $G = (A, B, E)$ is a bipartite permutation graph iff it admits an ordering of A that has the **adjacency** and **enclosure properties**

• • •

Basics…

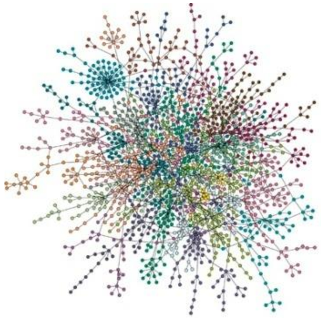


- **Graph Class: Bipartite Permutation Graphs**
  - A graph is a **bipartite permutation graph**, if it is both bipartite and a permutation graph.
  - A bipartite graph is a bipartite permutation graph iff it admits a **strong ordering**.
  - A bipartite graph $G = (A, B, E)$ is a bipartite permutation graph iff it admits an ordering of A that has the **adjacency** and **enclosure properties**

• • •

• • •

BASICS…



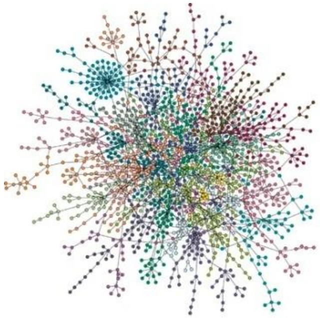
- **Graph Class: Bipartite Permutation Graphs**
  - A graph is a **bipartite permutation graph**, if it is both bipartite and a permutation graph.
  - A bipartite graph is a bipartite permutation graph iff it admits a **strong ordering**.
  - A bipartite graph $G = (A, B, E)$ is a bipartite permutation graph iff it admits an ordering of A that has the **adjacency** and **enclosure properties**

• • •

• • •

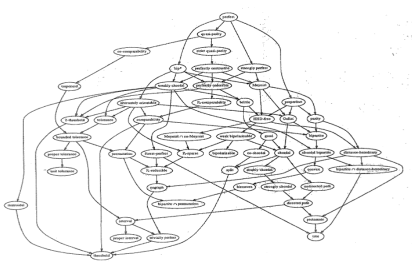
• • •

Basics…



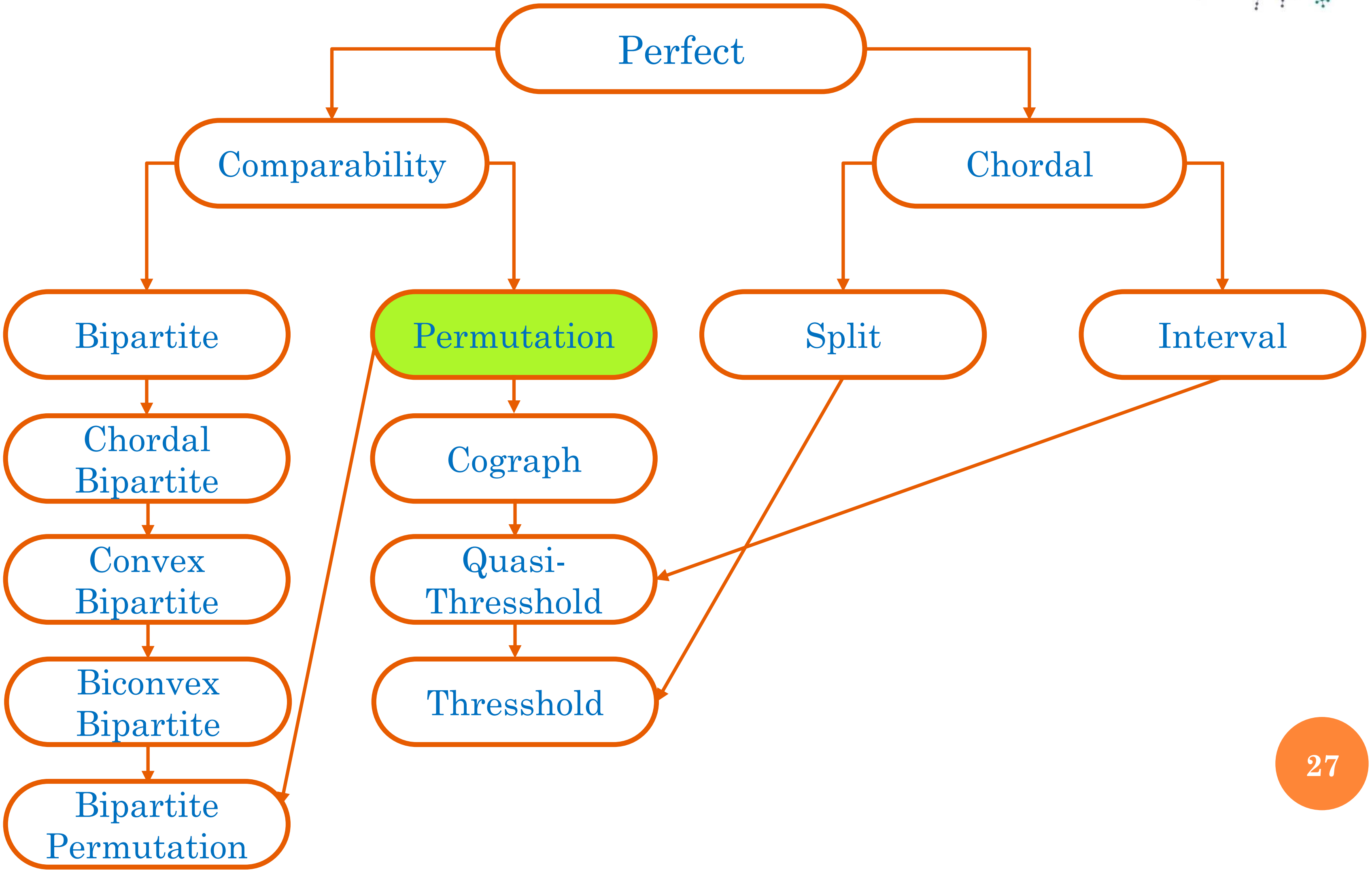
- **Graph Class: Bipartite Permutation Graphs**
  - A graph is a **bipartite permutation graph**, if it is both bipartite and a permutation graph.
  - A bipartite graph is a bipartite permutation graph iff it admits a **strong ordering**.
  - A bipartite graph $G = (A, B, E)$ is a bipartite permutation graph iff it admits an ordering of A that has the **adjacency** and **enclosure properties**

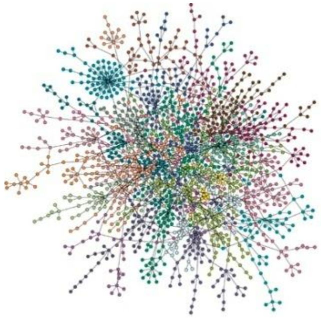
http://www.graphclasses.org/



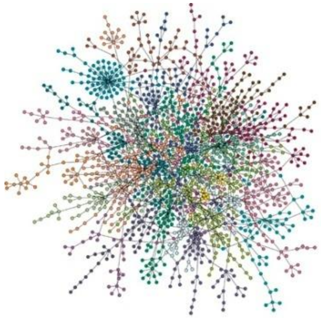
Basics…

- **Graph Classes**

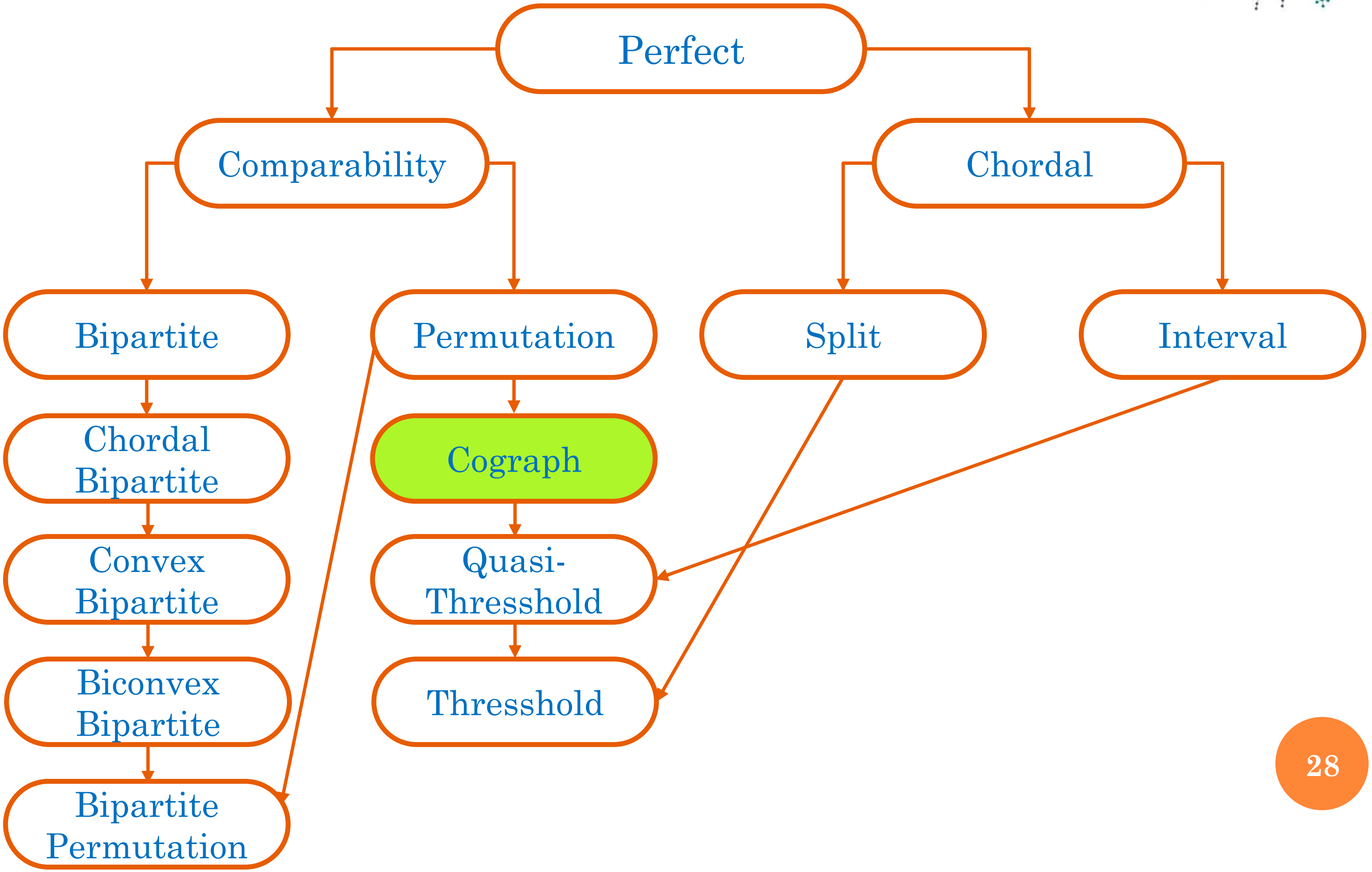




Basics…

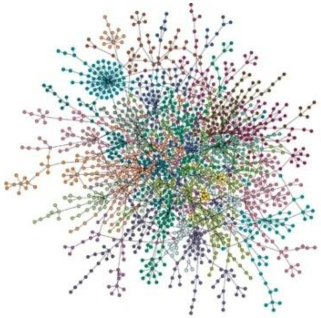


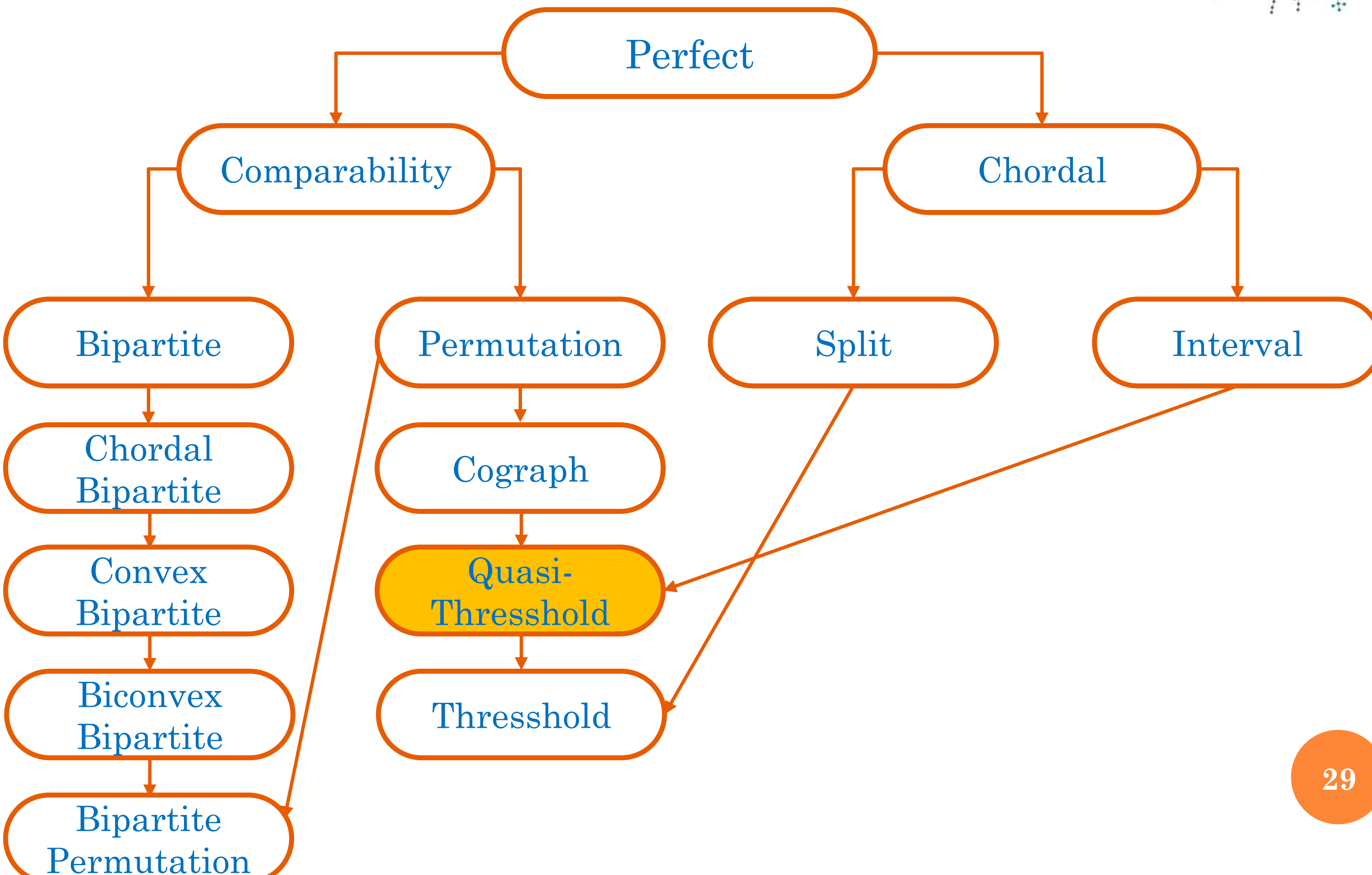
- **Graph Classes**



Basics…

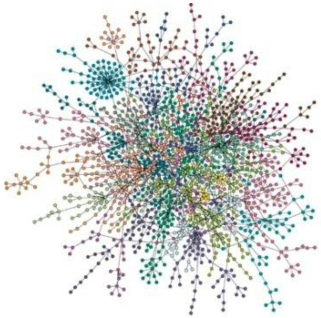
- **Graph Classes**

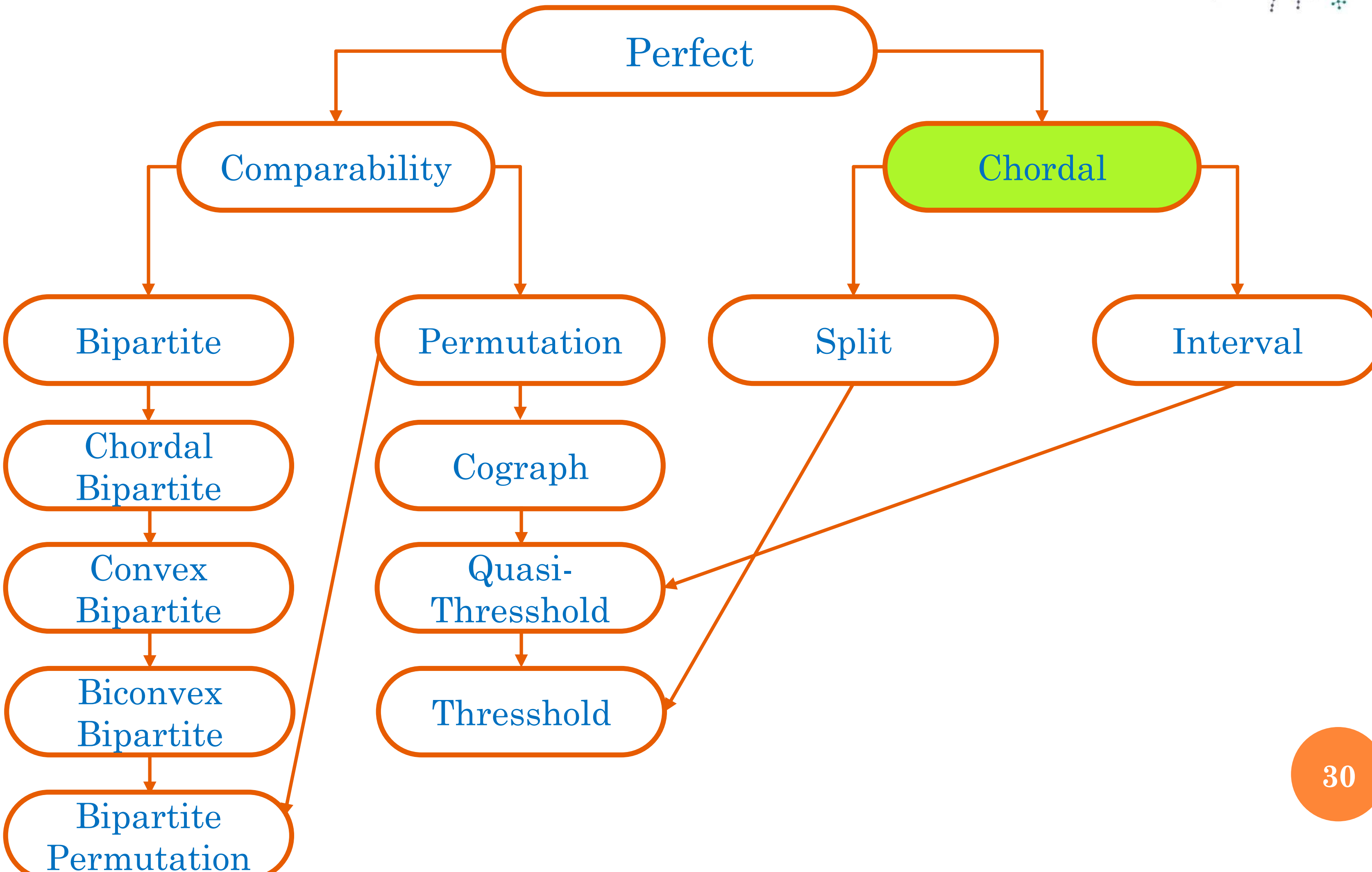




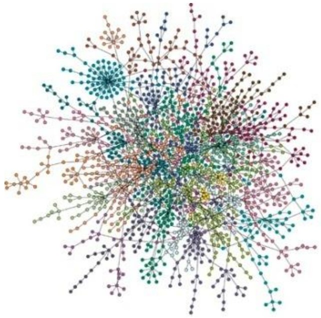
Basics…

- **Graph Classes**

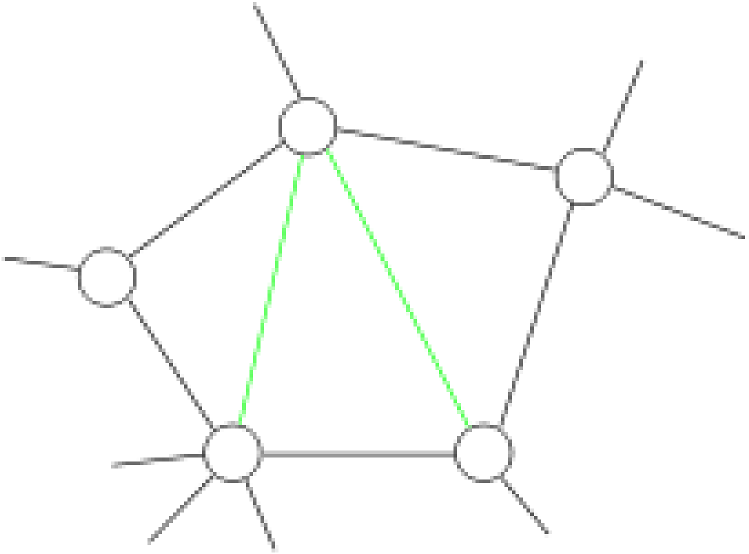




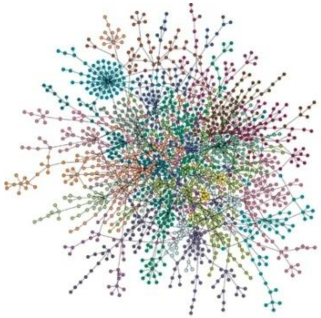
BASICS…



- **Graph Class: Chordal Graphs**
  - A **chordal graph** is one in which all cycles of four or more vertices have a chord, which is an edge that is not part of the cycle but connects two vertices of the cycle.
  - Every induced cycle in the graph should have exactly three vertices.
  - They are sometimes also called **triangulated graphs**.



Basics…

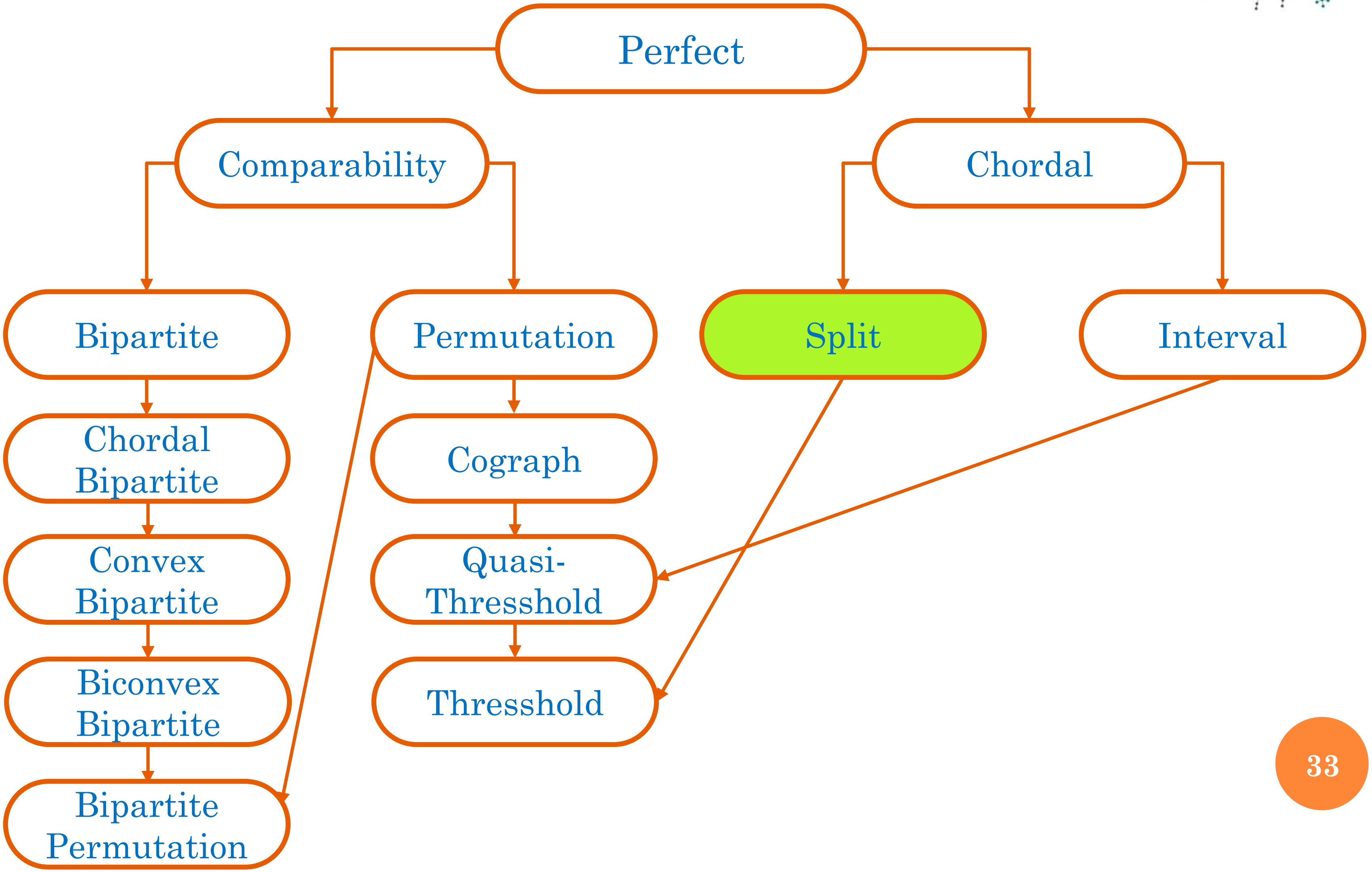


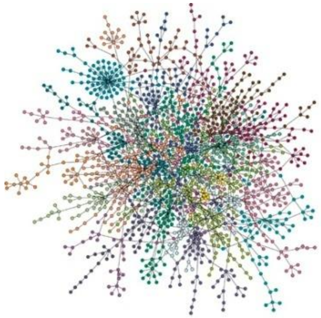
- **Graph Class: Chordal Graphs**
  - A **chordal graph** is one in which all cycles of four or more vertices have a chord, which is an edge that is not part of the cycle but connects two vertices of the cycle.
  - Every induced cycle in the graph should have exactly three vertices.
  - They are sometimes also called **triangulated graphs**.
  - **Transitive Orientation Property**
    - Each edge can be assigned a one-way direction in such a way that
    - the resulting oriented graph (V, F):

$$ab \in F \text{ and } bc \in F \Longrightarrow ac \in F \; (\forall\, a, b, c \in V)$$

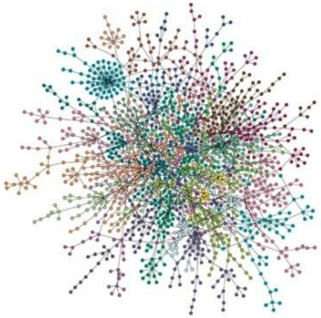
Basics…

- **Graph Classes**



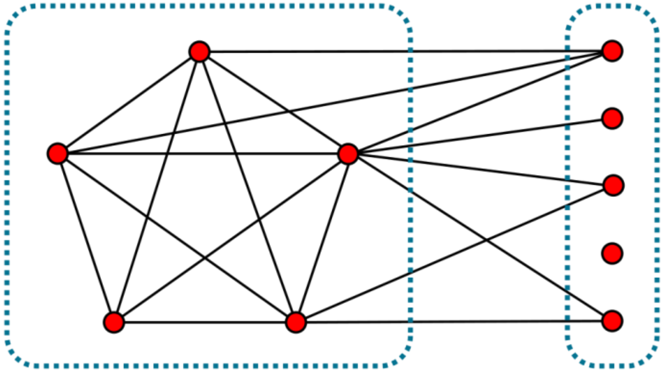


Basics…



Graph Class: Split Graphs

- A **split graph** is a graph in which the vertices can be partitioned into a clique and an independent set.
- A split graph may have more than one partition into a clique and an independent set.

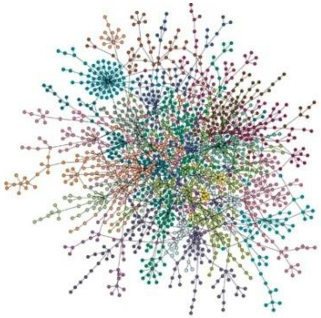


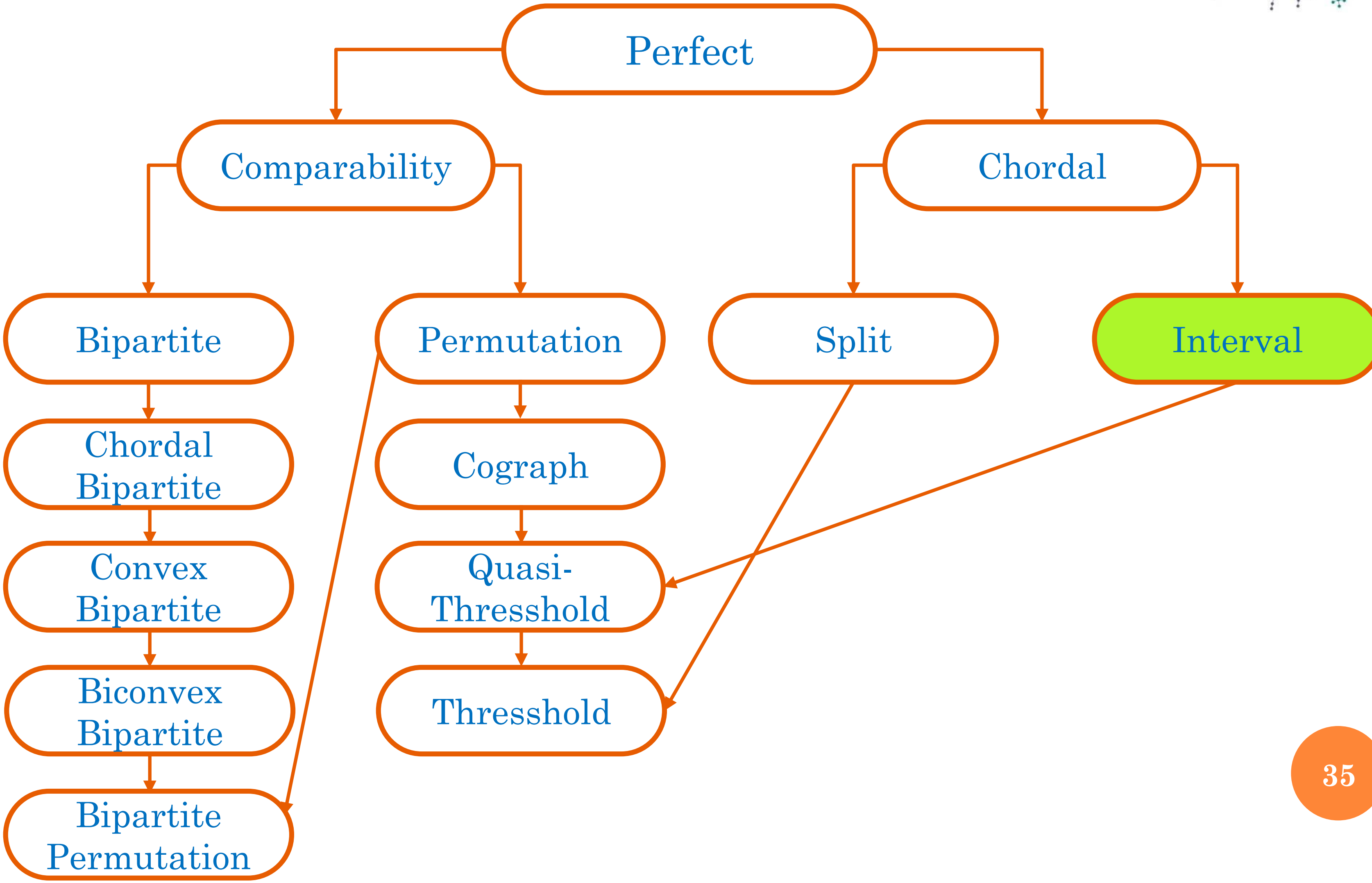
- Example: the path a–b–c is a split graph, the vertices of which can be partitioned in three different ways:

1. the clique {a,b} and the independent set {c}
2. the clique {b,c} and the independent set {a}
3. the clique {b} and the independent set {a,c}

Basics…

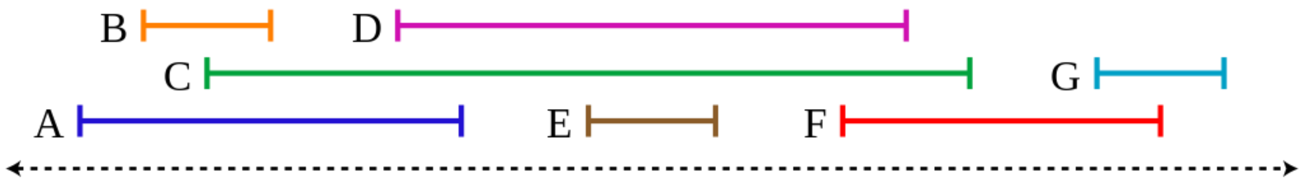
- **Graph Classes**



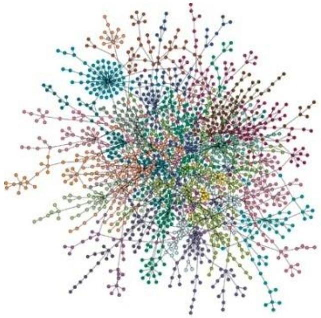


Basics…

- **Graph Class: Interval Graphs**
  - An **interval graph** is an undirected graph formed from a set of intervals on the real line, with a vertex for each interval and an edge between vertices whose intervals intersect.

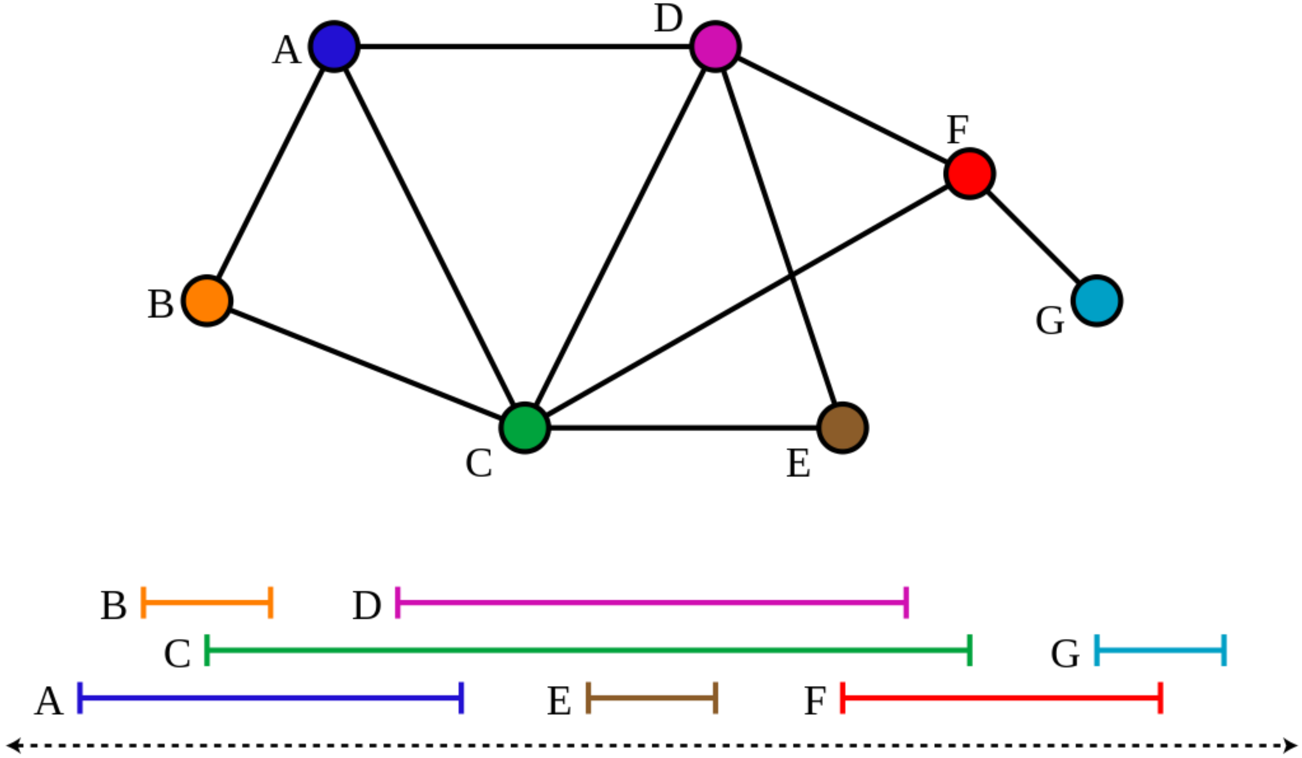


BASICS…

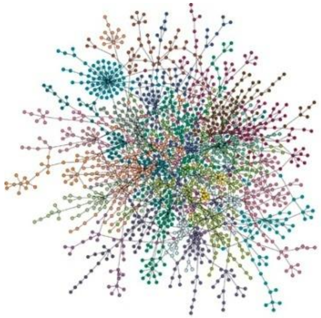


Graph Class: Interval Graphs

- An **interval graph** is an undirected graph formed from a set of intervals on the real line, with a vertex for each interval and an edge between vertices whose intervals intersect.



BASICS…



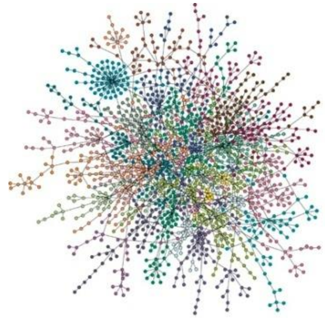
- **Graph Class: Interval Graphs**
  - An **interval graph** is an undirected graph formed from a set of intervals on the real line, with a vertex for each interval and an edge between vertices whose intervals intersect.
  - **Propositinon 1** : An induced subgraph of an interval graph is an interval graph.

    Proof.
    If $[I_V], v \in V$, is an interval representation of a graph $G = (V, E)$. Then, $[I_V], v \in X$, is an interval representation of the induced subgraph $G_X = (X, EX)$.

  - **Propositinon 2** : An interval graph satisfies the triangulated graph property.

    Proof.
    Suppose G contains a cordless cycle $[v_0, v_1, \ldots., v_{l\,-\,1}\ v_0]$ with $l > 3$.
    Let $I_K \rightarrow v_K$. For i =1, 2,…, l-1, choose a point $P_i \in {}'Ii_{\,-\,1} \cap I_i$.
    Since $I_{i\,-\,1}$ and $I_{i\,+\,1}$ do not overlap, the points P_i constitute a strictly increasing or decreasing sequence. Therefore, it is impossible for the intervals I_0 and $I_{l\,-\,1}$ to intersect, contradicting the criterion that $v_0, v_{l\,-\,1}$ is an edge of G.

BASICS…



- **Graph Class: Interval Graphs**
  - APPLICATION!

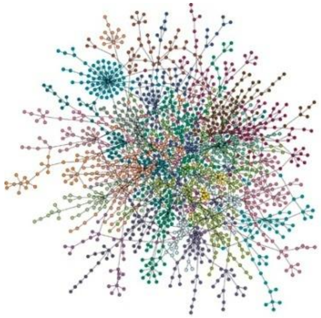
Let M a set of medicines $\{F_1, F_2, \dots, F_n\}$ $(n \geq 1)$, each one preserved in its own temperature range, let $[s_i, t_i]$, $1 \leq i \leq n$.

Design an algorithm that will compute the maximum number of medicines F_{max} that can be preserved over the minimum required temperature C_{min} .

F_1 = [4, 15]
F_2 = [3, 8]
F_3 = [0, 12]
F_4 = [5, 16]
F_5 = [1, 13],
F_6 = [11, 16]
F_7 = [2, 14]

$F_{max} = 6$ AND $C_{min} = 5$.

Basics…



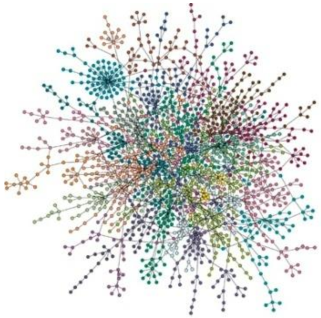
Graph Class: Interval Graphs

- APPLICATION!

  Let M a set of medicines $\{F_1, F_2, \ldots, F_n\}$ $(n \geq 1)$, each one preserved in its own temperature range, let $[s_i, t_i]$, $1 \leq i \leq n$.

  Design an algorithm that will compute the maximum number of medicines F_{max} that can be preserved over the minimum required temperature C_{min}.

BASICS…

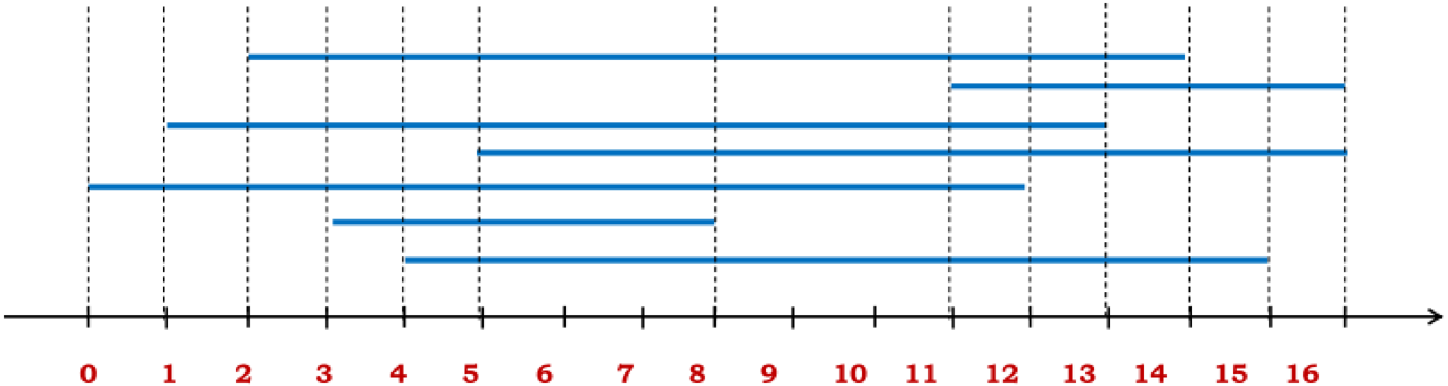


- **Graph Class: Interval Graphs**
  - APPLICATION!

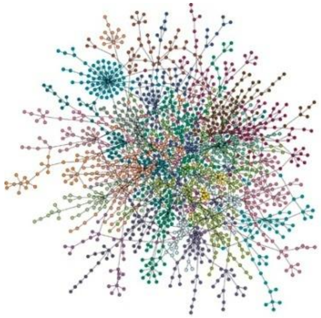
Let M a set of medicines $\{F_1, F_2, \ldots, F_n\}$ $(n \geq 1)$, each one preserved in its own temperature range, let $[s_i, t_i]$, $1 \leq i \leq n$.

Design an algorithm that will compute the maximum number of medicines F_{max} that can be preserved over the minimum required temperature C_{min} .

$F_1 = [4, 15]; F_2 = [3, 8]; F_3 = [0, 12]; F_4 = [5, 16];$
$F_5 = [1, 13]; F_6 = [11, 16]; F_7 = [2, 14]$



BASICS…

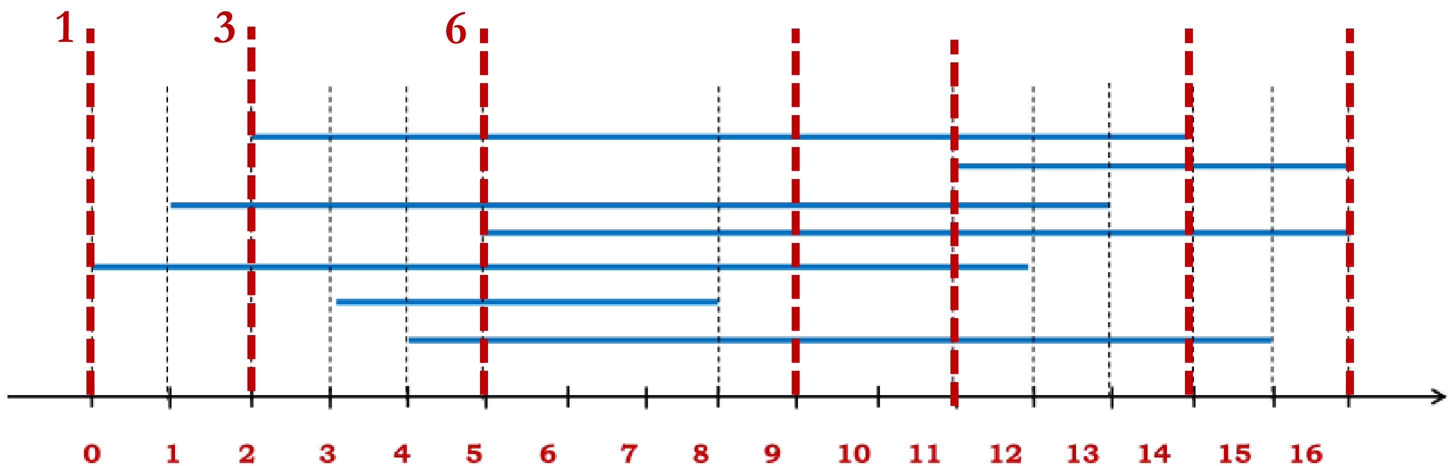


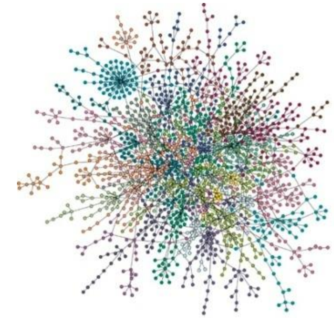
- **Graph Class: Interval Graphs**
  - APPLICATION!

Let M a set of medicines $\{F_1, F_2, \dots, F_n\}$ $(n \geq 1)$, each one preserved in its own temperature range, let $[s_i, t_i]$, $1 \leq i \leq n$.

Design an algorithm that will compute the maximum number of medicines F_{max} that can be preserved over the minimum required temperature C_{min} .

$F_1 = [4, 15]$; $F_2 = [3, 8]$; $F_3 = [0, 12]$; $F_4 = [5, 16]$;
$F_5 = [1, 13]$; $F_6 = [11, 16]$; $F_7 = [2, 14]$





Basics…

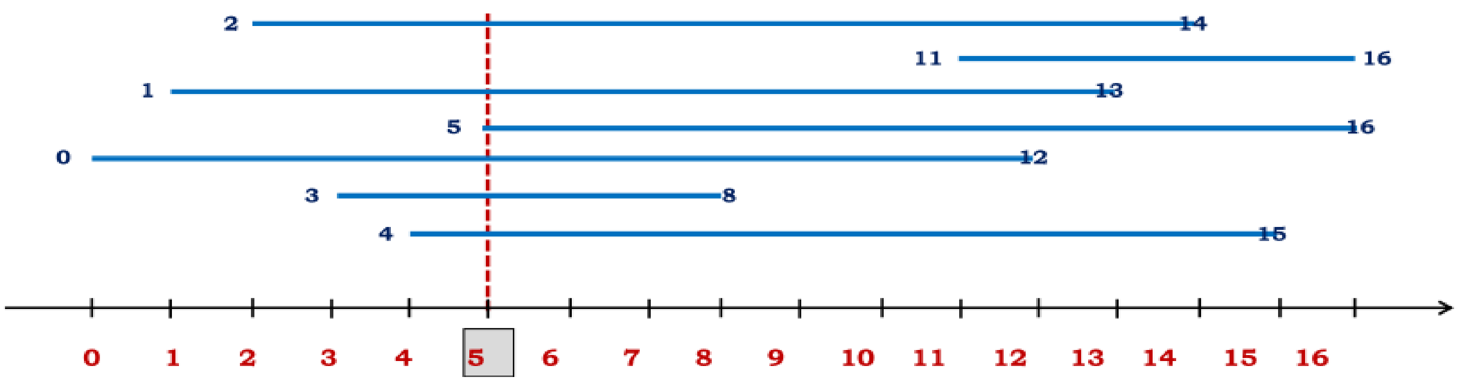
- **Graph Class: Interval Graphs**
  - APPLICATION!

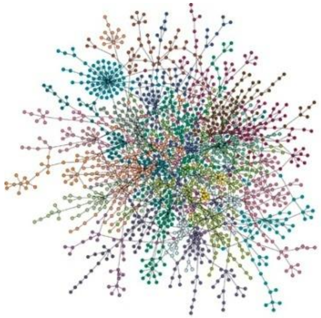
Let M a set of medicines $\{F_1, F_2, \ldots, F_n\}$ $(n \geq 1)$, each one preserved in its own temperature range, let $[s_i, t_i]$, $1 \leq i \leq n$.

Design an algorithm that will compute the maximum number of medicines F_{max} that can be preserved over the minimum required temperature C_{min}.

$F_1 = [4, 15]$; $F_2 = [3, 8]$; $F_3 = [0, 12]$; $F_4 = [5, 16]$;
$F_5 = [1, 13]$; $F_6 = [11, 16]$; $F_7 = [2, 14]$

$F_{max} = 6$ AND $C_{min} = 5$.



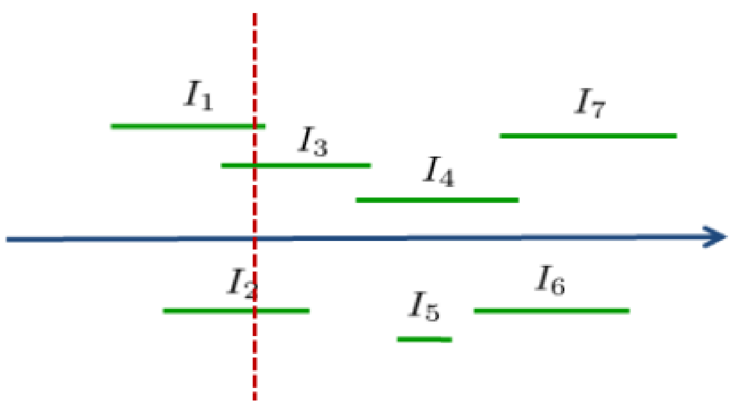


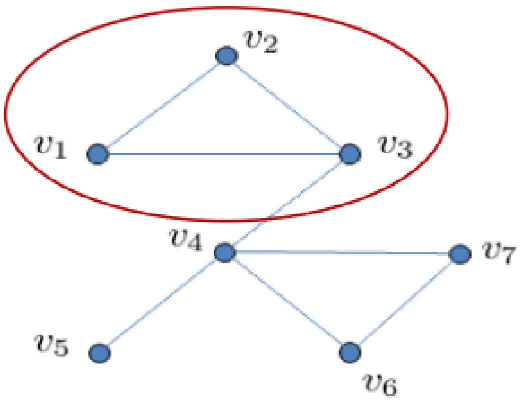
Basics…

- **Graph Class: Interval Graphs**
  - APPLICATION!

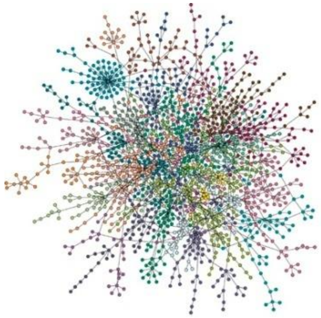
    Let M a set of medicines $\{F_1, F_2, \dots, F_n\}$ $(n \geq 1)$, each one preserved in its own temperature range, let $[s_i, t_i]$, $1 \leq i \leq n$.

    Design an algorithm that will compute the maximum number of medicines F_{max} that can be preserved over the minimum required temperature C_{min} .



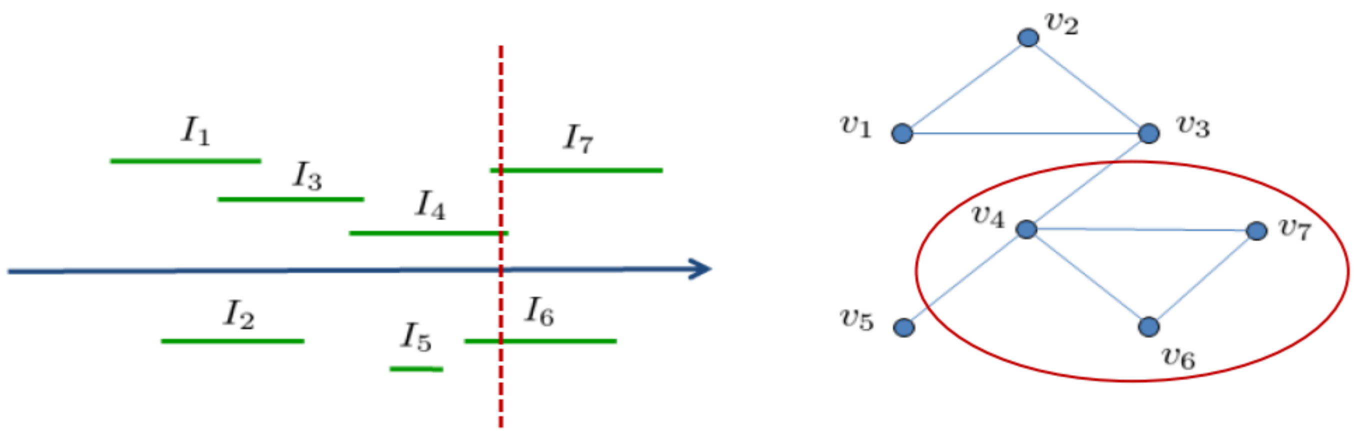


BASICS…

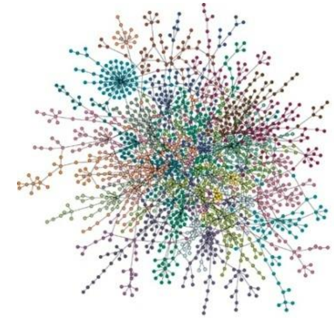


- **Graph Class: Interval Graphs**
  - APPLICATION!

    The manipulation of such problems over interval graphs is made through the utilization of the computation of the maximum clique.

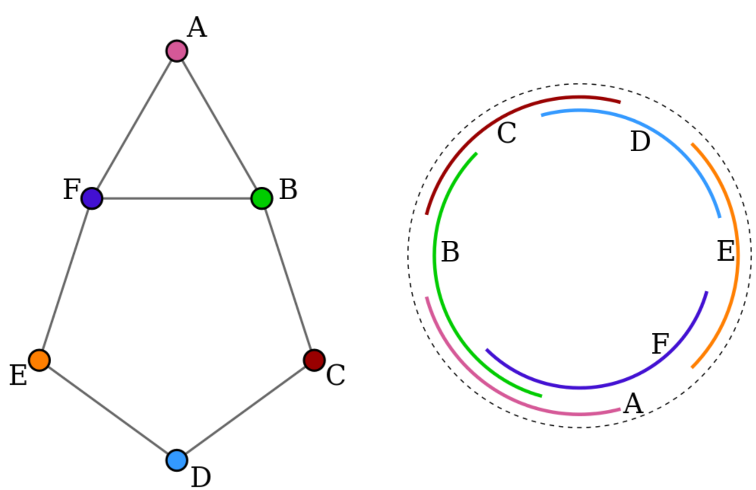


Basics…

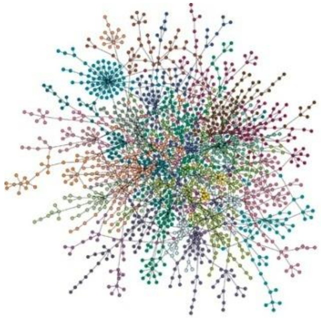


Graph Class: Circular-Arc Graphs

- A **Circular-arc graph** is the intersection graph of a set of arcs on the circle.
- It has one vertex for each arc in the set, and an edge between every pair of vertices corresponding to arcs that intersect
- If a circular-arc graph G has an arc model that leaves some point of the circle uncovered, the circle can be cut at that point and stretched to a line, which results in an interval representation.
- Unlike interval graphs, however, circular-arc graphs are not always perfect, as the odd chordless cycles C_5, C_7, etc., are circular-arc graphs.

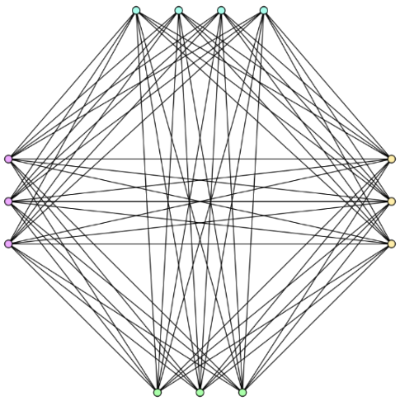


BASICS…



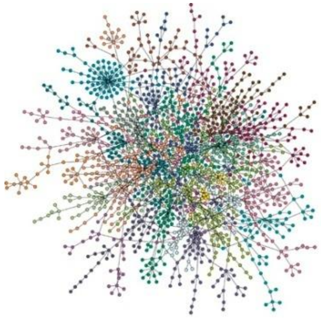
- **Graph Class: Cographs Graphs**
  - A **cograph**, or complement-reducible graph, or P_4-free graph, is a graph that can be generated from the single-vertex graph K_1 by complementation and disjoint union.
  - The family of cographs is the smallest class of graphs that includes K_1 and is closed under complementation and disjoint union
  - The cographs may be defined as the graphs that can be constructed using the following operations, starting from the single-vertex graph:

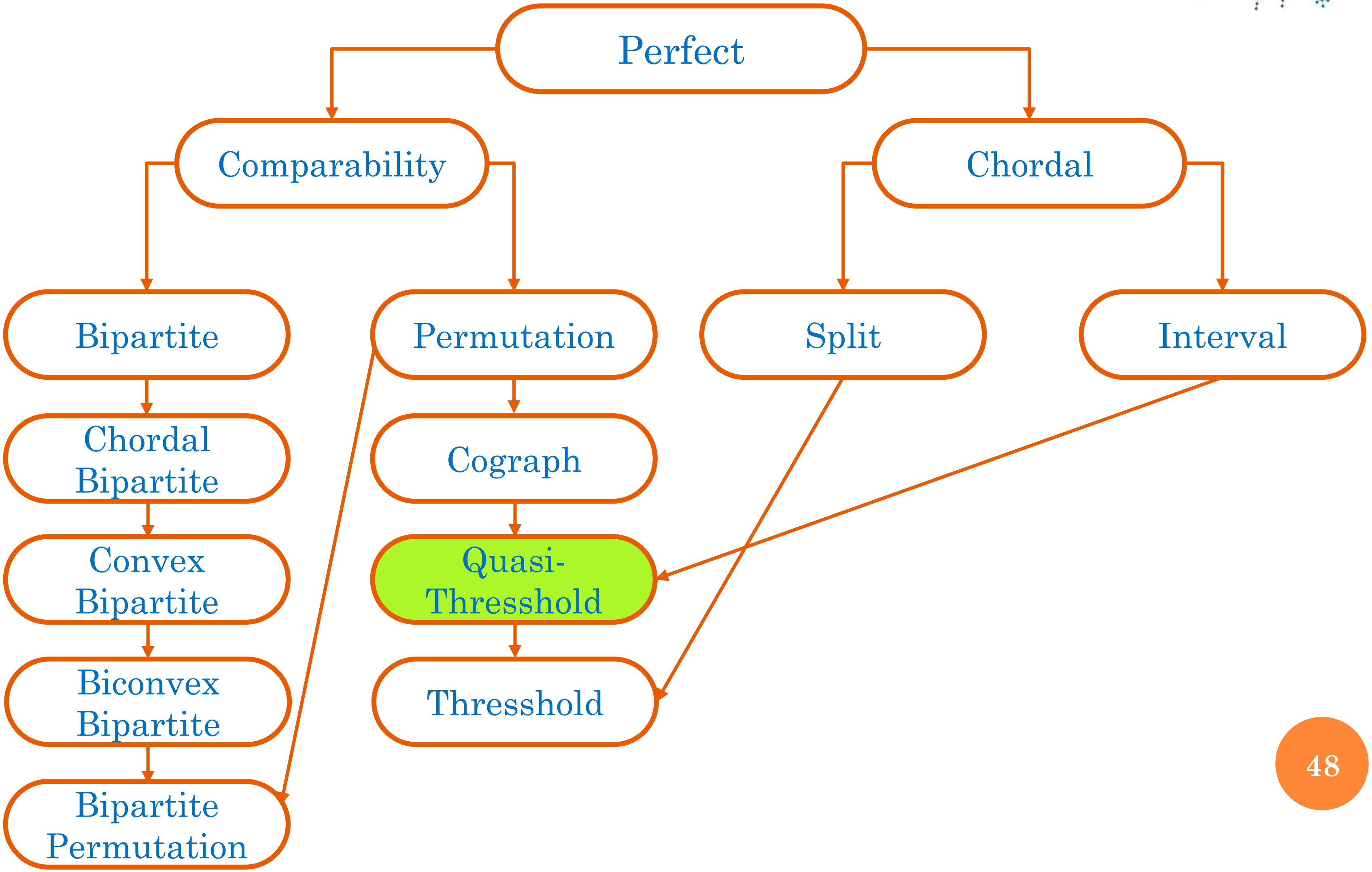
    1. any single vertex graph,
       … is a cograph;
    2. if G is a cograph,
       … so is its complement graph $\bar{G}$;
    3. if G and H are cographs,
       … so is their disjoint union $G \cup H$.



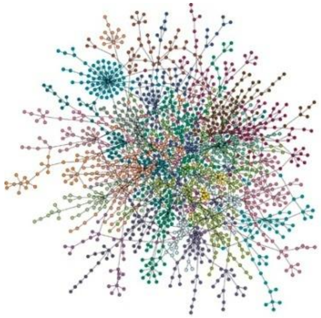
Basics…

- **Graph Classes**





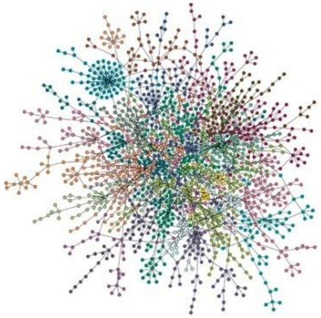
Basics…



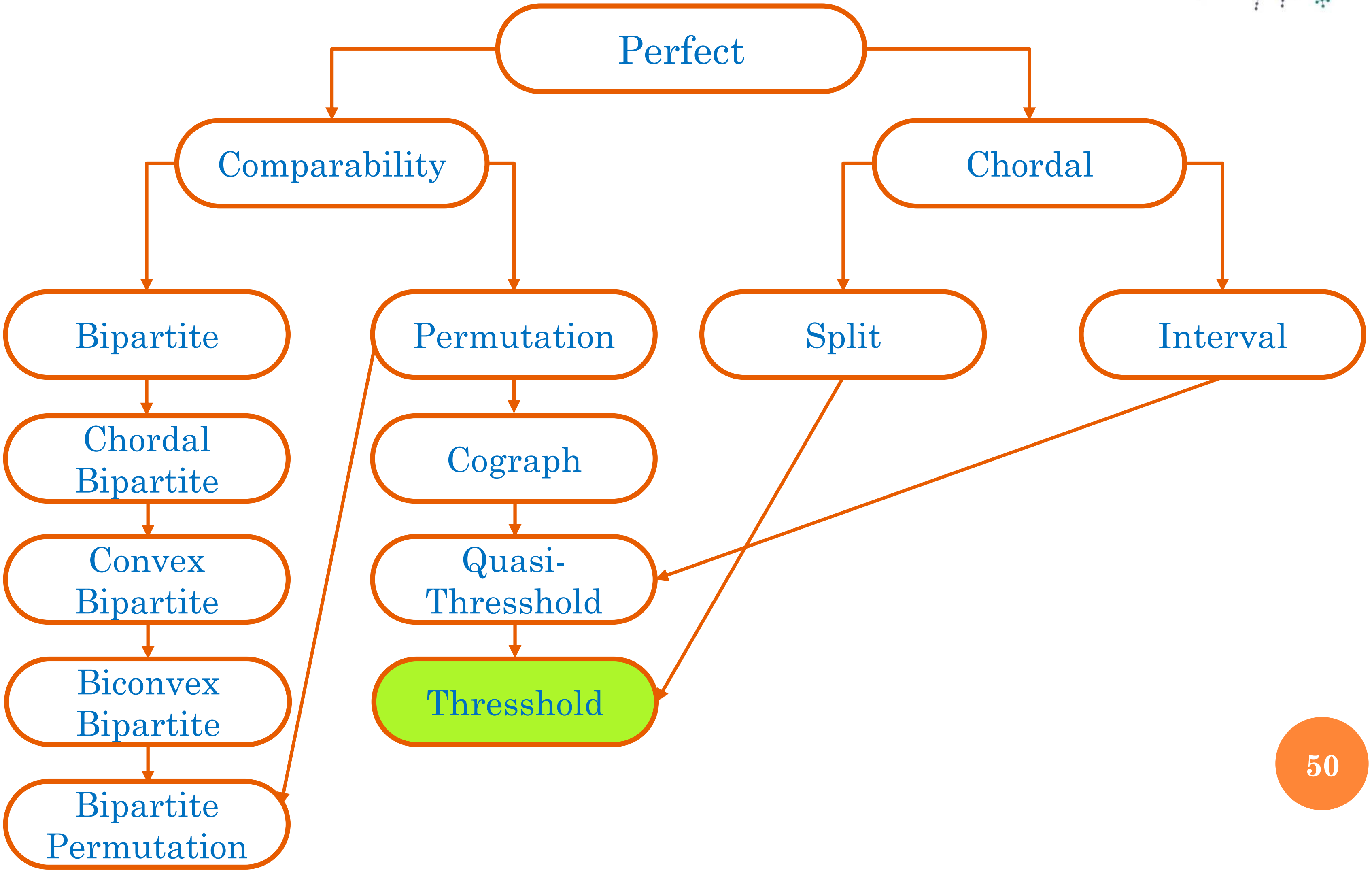
- **Graph Class: Threshold Graphs**
  - The **quasi-threshold graphs** are defined recursively as follows:.
    1. K_1 is a quasi-threshold graph
    2. Adding a new vertex adjacent to all vertices of a quasi-threshold graph results in a quasi-threshold graph
    3. The disjoint union of two quasi-threshold graphs results in a quasi-threshold graph..

…

Basics…

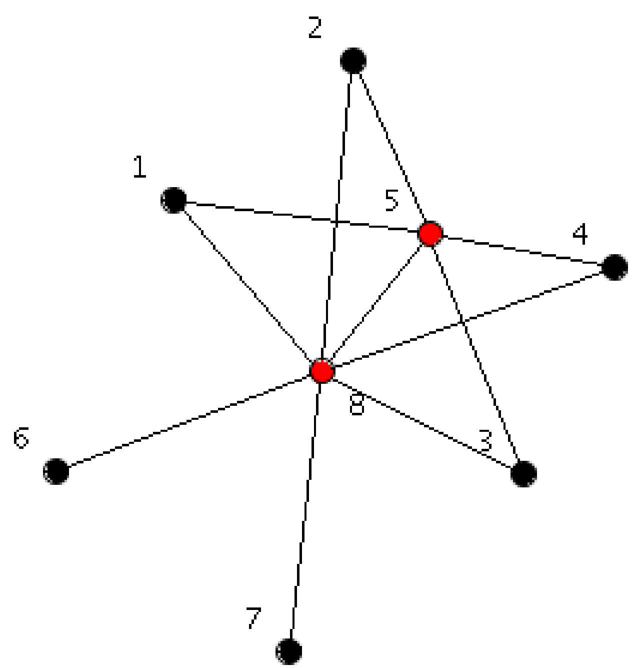


- **Graph Classes**

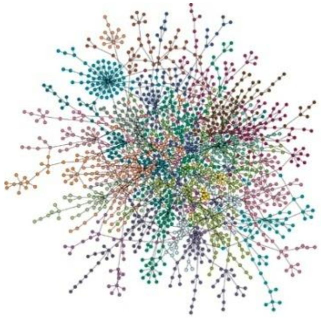


Basics…

- **Graph Class: Threshold Graphs**
  - A **threshold graph** is a graph that can be constructed from a one-vertex graph by repeated applications of the following two operations:
    - Addition of a single isolated vertex to the graph.
    - Addition of a single dominating vertex to the graph, i.e. a single vertex that is connected to all other vertices.



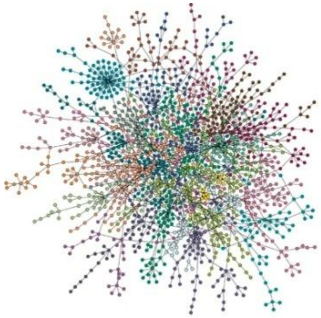
Basics…



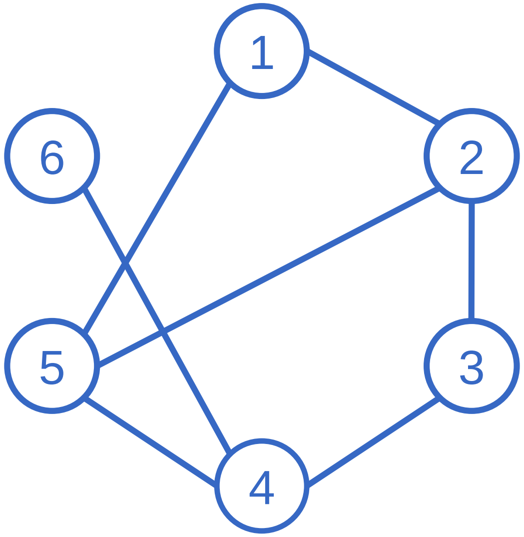
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **graph** G is represented by a $n \times n$ matrix, let A, where $\forall\ i, j\ \in\ V(G)\ \rightarrow Aij = 1\ if\ (i, j)\ \in\ E(G)$.

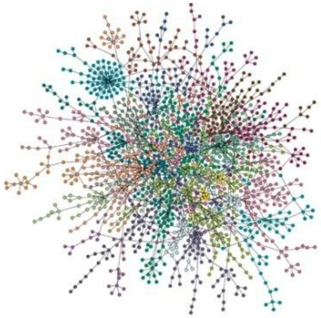
Basics…



- **Graph Representation – Adjacency Matrix**
  - Let undirected graph, G = (V, E):
    - V = vertices - $V(G)$,
    - E = edges between pairs of vertices - $E(G)$,
    - Size parameters: $n = |V|, m = |E|$.
  - A **graph** G is represented by a $n \times n$ matrix, let A, where $\forall\ i, j \in V(G) \rightarrow Aij = 1\ if\ (i, j) \in E(G)$.

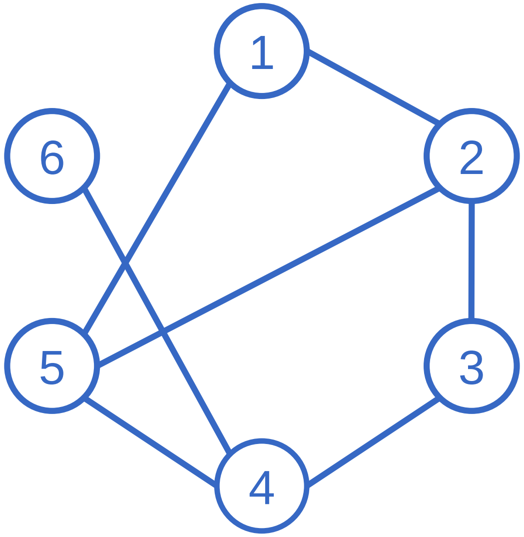


Basics…



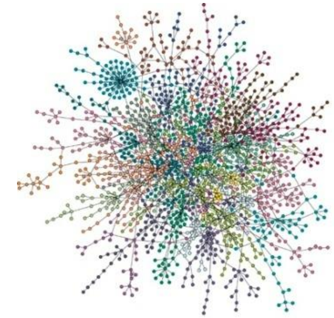
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **graph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow Aij = 1\; if\; (i, j) \in E(G)$.



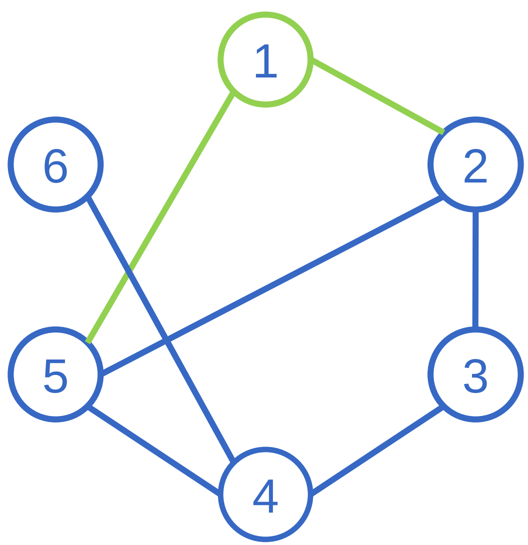
	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Basics…



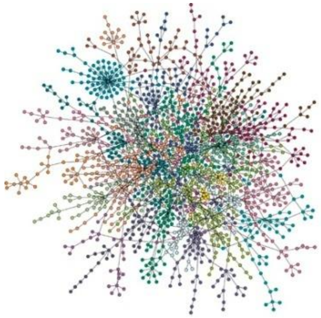
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **graph** G is represented by a $n \times n$ matrix, let A, where $\forall\ i, j \in V(G) \rightarrow Aij = 1\ if\ (i, j) \in E(G)$.



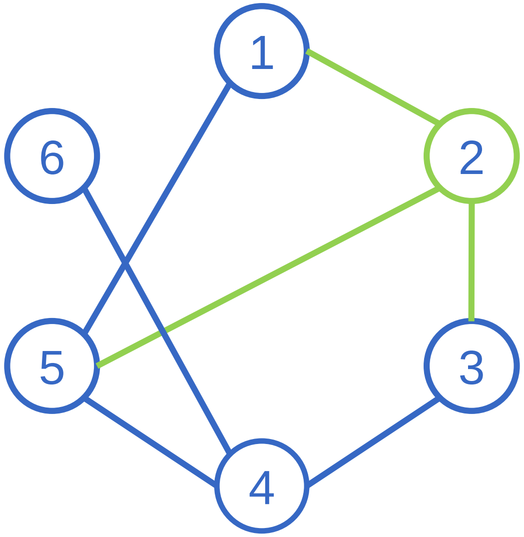
	1	2	3	4	5	6
1	0	1	0	0	1	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Basics…



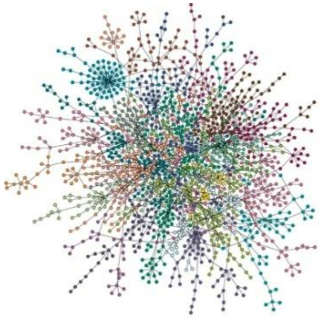
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **graph** G is represented by a $n \times n$ matrix, let A, where $\forall\ i, j\ \in\ V(G)\ \rightarrow Aij = 1\ if\ (i, j)\ \in\ E(G)$.



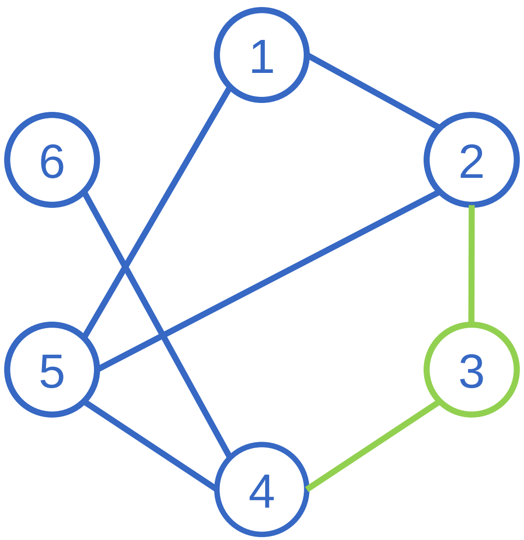
	1	**2**	**3**	**4**	**5**	**6**
1	0	**1**	0	0	**1**	0
2	**1**	0	**1**	0	**1**	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Basics…



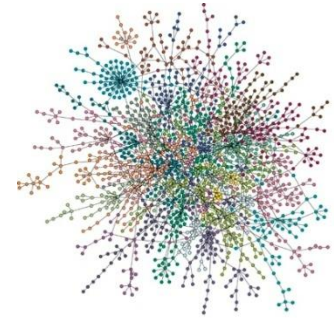
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **graph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow Aij = 1\; if\; (i, j) \in E(G)$.



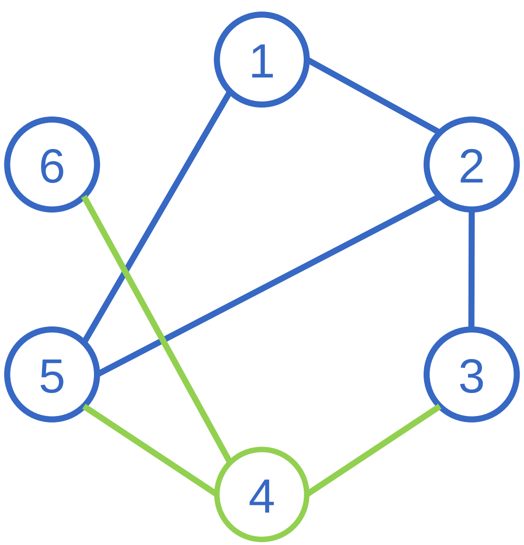
	1	2	3	4	5	6
1	0	1	0	0	1	0
2	1	0	1	0	1	0
3	0	1	0	1	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Basics…



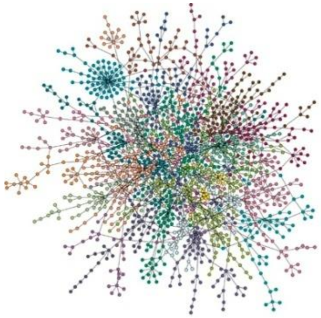
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **graph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow Aij = 1\; if\; (i, j) \in E(G)$.



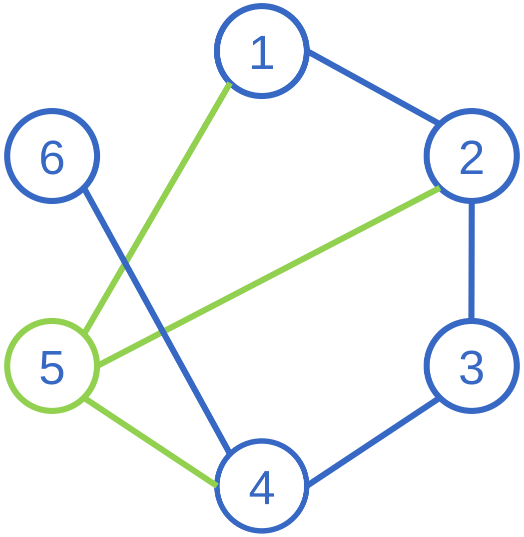
	1	2	3	4	5	6
1	0	1	0	0	1	0
2	1	0	1	0	1	0
3	0	1	0	1	0	0
4	0	0	1	0	1	1
5	0	0	0	0	0	0
6	0	0	0	0	0	0

BASICS…



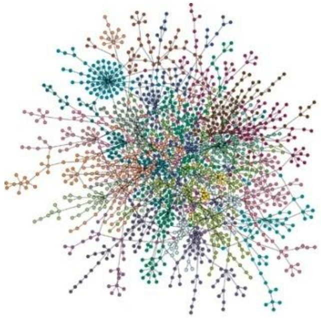
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **graph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow Aij = 1\; if\; (i, j) \in E(G)$.

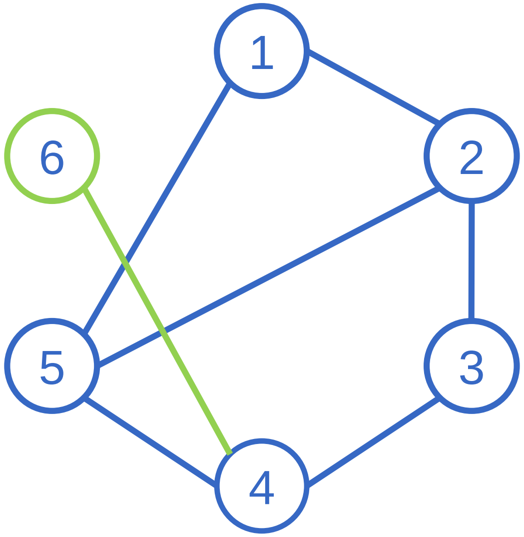


	1	2	3	4	5	6
1	0	**1**	0	0	**1**	0
2	**1**	0	**1**	0	**1**	0
3	0	**1**	0	**1**	0	0
4	0	0	**1**	0	**1**	**1**
5	**1**	**1**	0	**1**	0	0
6	0	0	0	0	0	0

Basics…

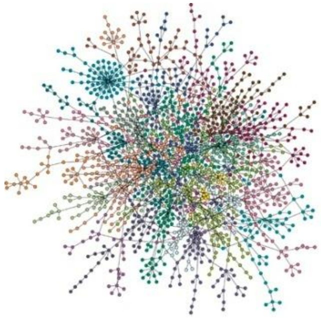


- **Graph Representation – Adjacency Matrix**
  - Let undirected graph, G = (V, E):
    - V = vertices - $V(G)$,
    - E = edges between pairs of vertices - $E(G)$,
    - Size parameters: $n = |V|, m = |E|$.
  - A **graph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow Aij = 1\ if\ (i, j) \in E(G)$.



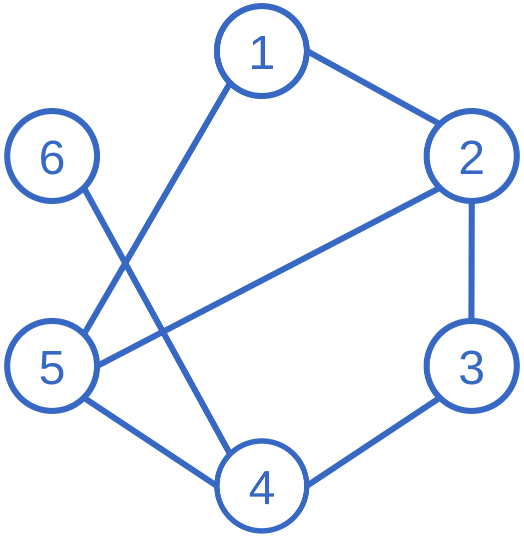
	1	2	3	4	5	6
1	0	1	0	0	1	0
2	1	0	1	0	1	0
3	0	1	0	1	0	0
4	0	0	1	0	1	1
5	1	1	0	1	0	0
6	0	0	0	1	0	0

Basics…



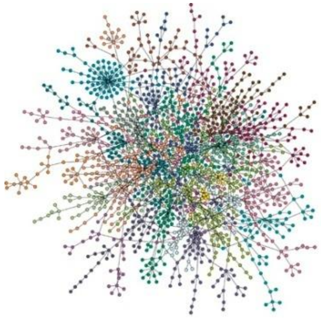
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **graph** G is represented by a $n \times n$ matrix, let A, where $\forall\ i, j \in V(G) \rightarrow Aij = 1\ if\ (i, j) \in E(G)$.



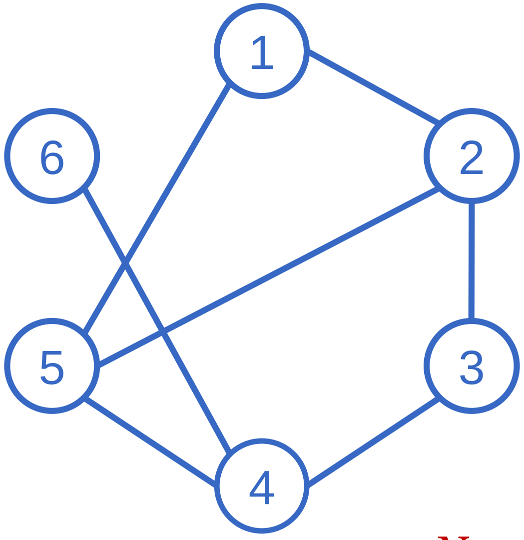
	1	2	3	4	5	6
1	0	1	0	0	1	0
2	1	0	1	0	1	0
3	0	1	0	1	0	0
4	0	0	1	0	1	1
5	1	1	0	1	0	0
6	0	0	0	1	0	0

Basics…



Graph Representation – Adjacency Matrix

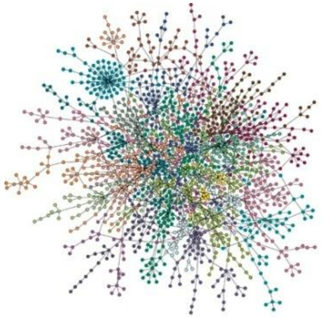
- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **graph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow Aij = 1\ if\ (i, j) \in E(G)$.



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

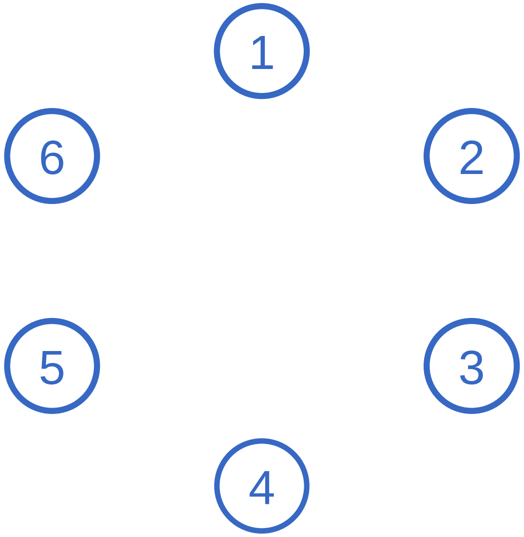
No self-loops encountered !!!

Basics…

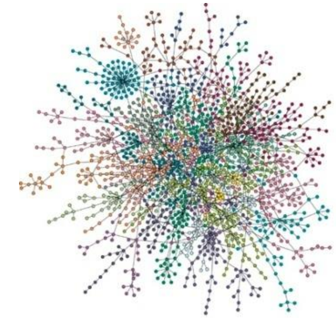


Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **graph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow Aij = 1\, if\ (i, j) \in E(G)$.



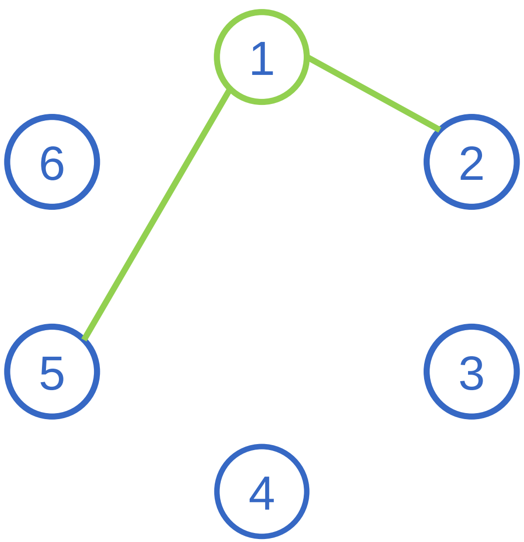
	1	**2**	**3**	**4**	**5**	**6**
1	0	**1**	0	0	**1**	0
2	**1**	0	**1**	0	**1**	0
3	0	**1**	0	**1**	0	0
4	0	0	**1**	0	**1**	**1**
5	**1**	**1**	0	**1**	0	0
6	0	0	0	**1**	0	0



BASICS…

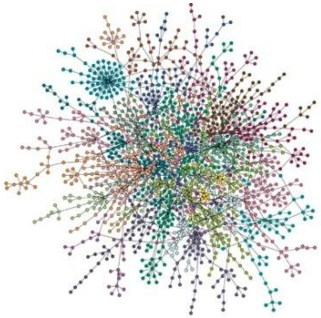
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **graph** G is represented by a $n \times n$ matrix, let A, where $\forall\ i, j \ \in \ V(G) \ \rightarrow Aij = 1\ if\ (i, j) \ \in \ E(G)$.



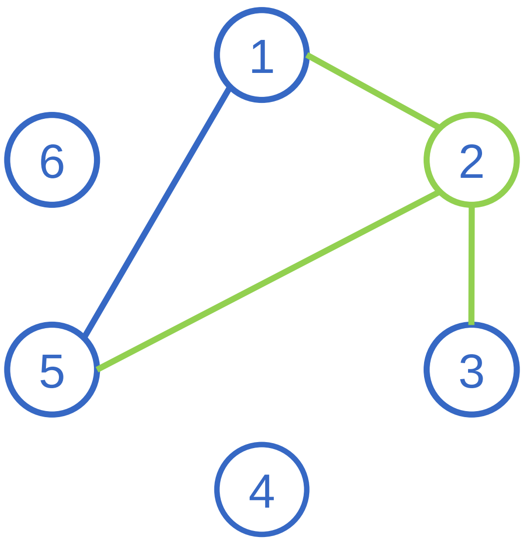
	1	2	3	4	5	6
1	0	1	0	0	1	0
2	1	0	1	0	1	0
3	0	1	0	1	0	0
4	0	0	1	0	1	1
5	1	1	0	1	0	0
6	0	0	0	1	0	0

Basics…



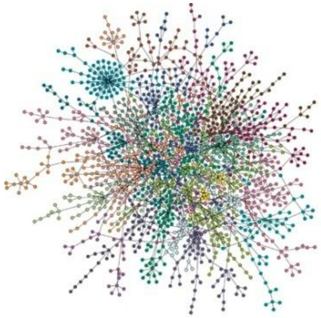
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **graph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow Aij = 1\ if\ (i, j) \in E(G)$.

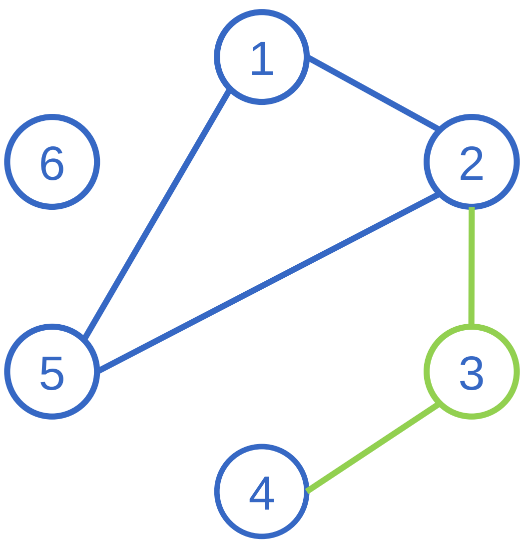


	1	2	3	4	5	6
1	0	1	0	0	1	0
2	1	0	1	0	1	0
3	0	1	0	1	0	0
4	0	0	1	0	1	1
5	1	1	0	1	0	0
6	0	0	0	1	0	0

Basics…

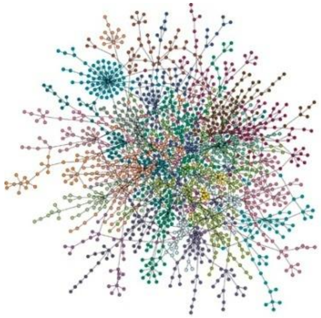


- **Graph Representation – Adjacency Matrix**
  - Let undirected graph, G = (V, E):
    - V = vertices - $V(G)$,
    - E = edges between pairs of vertices - $E(G)$,
    - Size parameters: $n = |V|, m = |E|$.
  - A **graph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow Aij = 1\; if\; (i, j) \in E(G)$.



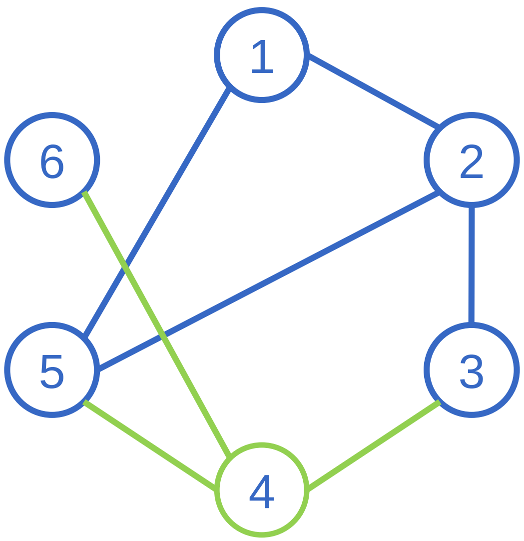
	1	2	3	4	5	6
1	0	**1**	0	0	**1**	0
2	**1**	0	**1**	0	**1**	0
3	0	**1**	0	**1**	0	0
4	0	0	**1**	0	**1**	**1**
5	**1**	**1**	0	**1**	0	0
6	0	0	0	**1**	0	0

Basics…



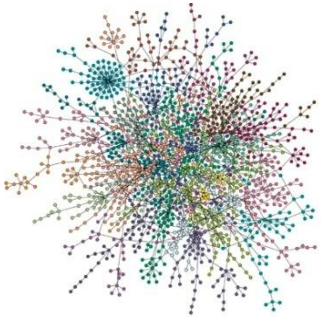
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **graph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i,j \in V(G) \rightarrow Aij = 1\ if\ (i,j) \in E(G)$.



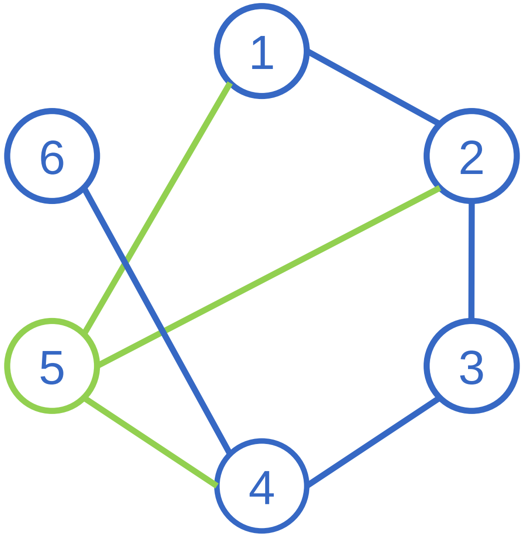
	1	2	3	4	5	6
1	0	1	0	0	1	0
2	1	0	1	0	1	0
3	0	1	0	1	0	0
4	0	0	1	0	1	1
5	1	1	0	1	0	0
6	0	0	0	1	0	0

Basics…



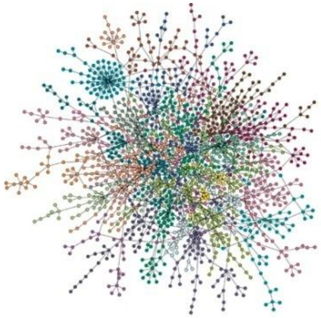
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **graph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow Aij = 1\; if\; (i, j) \in E(G)$.



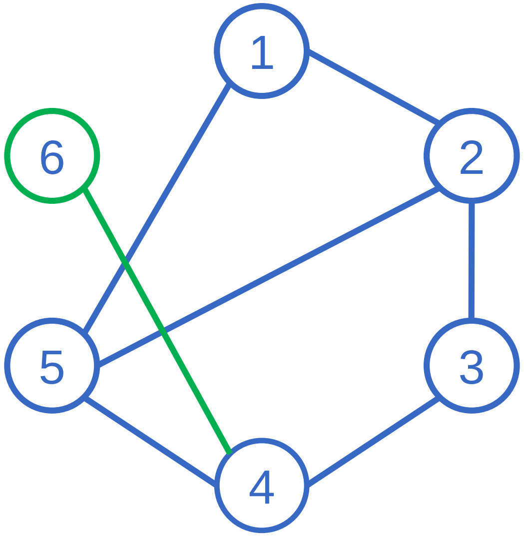
	1	2	3	4	5	6
1	0	1	0	0	1	0
2	1	0	1	0	1	0
3	0	1	0	1	0	0
4	0	0	1	0	1	1
5	1	1	0	1	0	0
6	0	0	0	1	0	0

BASICS…



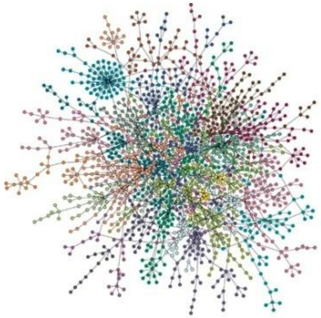
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **graph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow Aij = 1\; if\; (i, j) \in E(G)$.



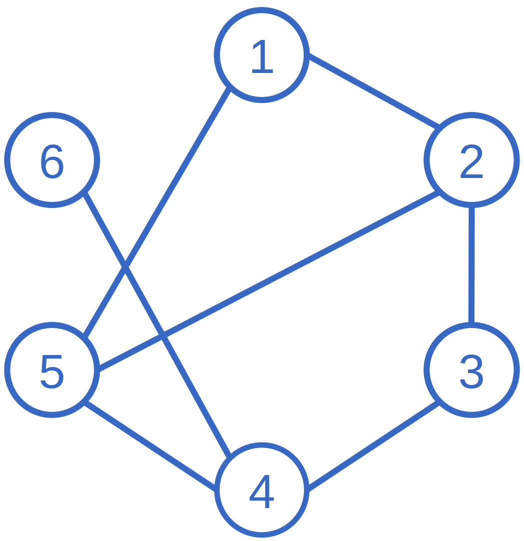
	1	2	3	4	5	6
1	0	1	0	0	1	0
2	1	0	1	0	1	0
3	0	1	0	1	0	0
4	0	0	1	0	1	1
5	1	1	0	1	0	0
6	0	0	0	1	0	0

Basics…



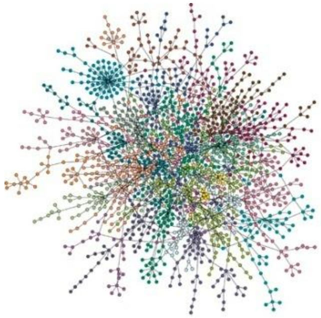
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **graph** G is represented by a $n \times n$ matrix, let A, where $\forall\ i, j\ \in\ V(G)\ \rightarrow A_{ij} = 1\ if\ (i, j)\ \in\ E(G)$.

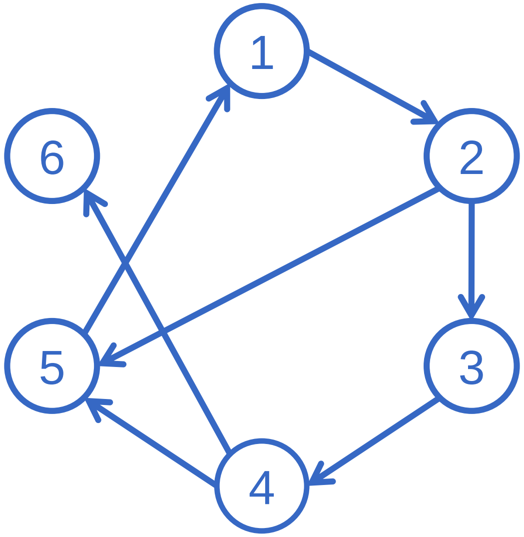


	1	**2**	**3**	**4**	**5**	**6**
1	0	**1**	0	0	**1**	0
2	**1**	0	**1**	0	**1**	0
3	0	**1**	0	**1**	0	0
4	0	0	**1**	0	**1**	**1**
5	**1**	**1**	0	**1**	0	0
6	0	0	0	**1**	0	0

Basics…



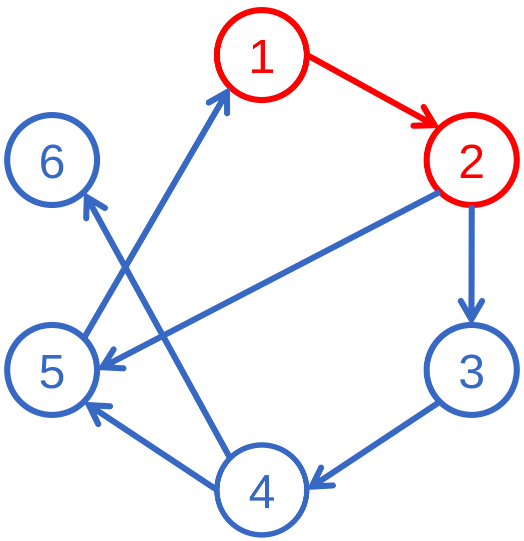
- **Graph Representation – Adjacency Matrix**
  - Let undirected graph, G = (V, E):
    - V = vertices - $V(G)$,
    - E = edges between pairs of vertices - $E(G)$,
    - Size parameters: $n = |V|, m = |E|$.
  - A **digraph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow A_{ij} = 1\ if\ (i, j) \in E(G)$.



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

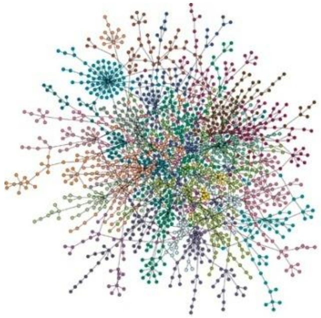
Basics…

- **Graph Representation – Adjacency Matrix**
  - Let undirected graph, G = (V, E):
    - V = vertices - $V(G)$,
    - E = edges between pairs of vertices - $E(G)$,
    - Size parameters: $n = |V|, m = |E|$.
  - A **digraph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow A_{ij} = 1\ if\ (i, j) \in E(G)$.



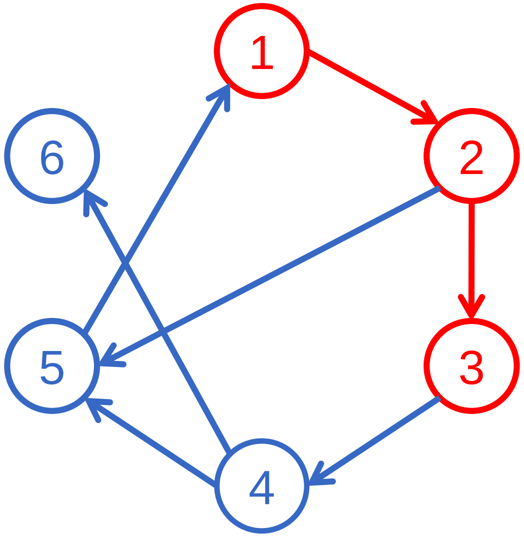
	1	2	3	4	5	6
1	0	1	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

BASICS…



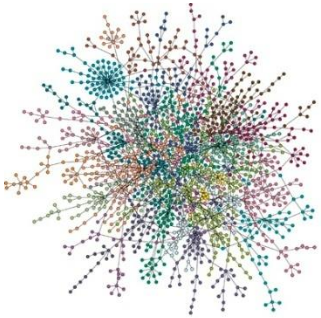
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **digraph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow A_{ij} = 1\ if\ (i, j) \in E(G)$.



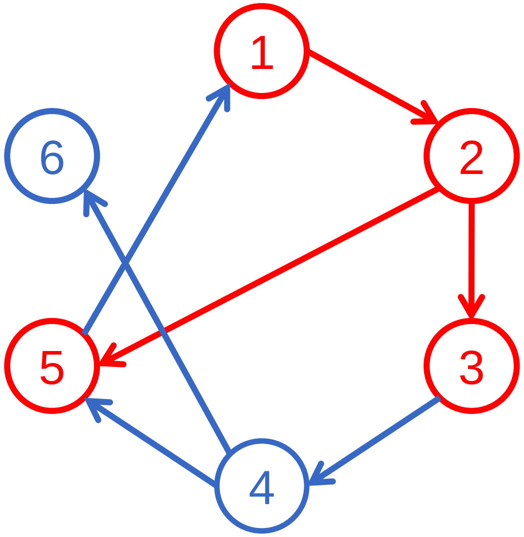
	1	2	3	4	5	6
1	0	**1**	0	0	0	0
2	0	0	**1**	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Basics…



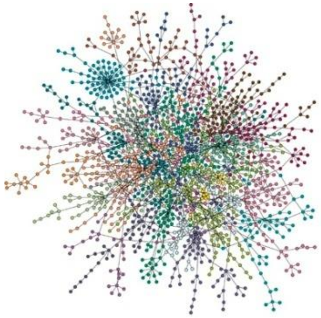
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **digraph** G is represented by a $n \times n$ matrix, let A, where $\forall\ i, j\ \in\ V(G)\ \rightarrow A_{ij} = 1\ if\ (i, j)\ \in\ E(G)$.



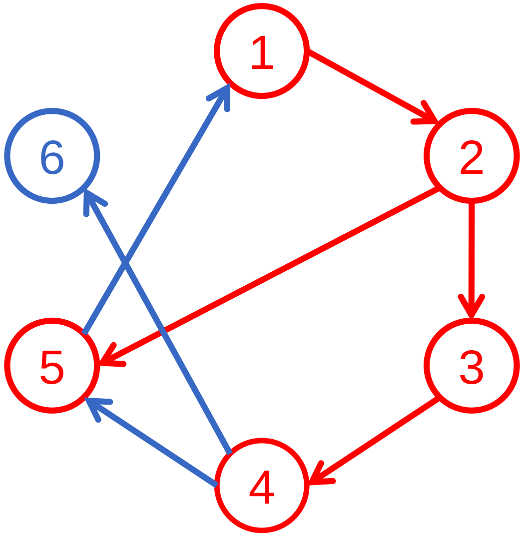
	1	2	3	4	5	6
1	0	1	0	0	0	0
2	0	0	1	0	1	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Basics…



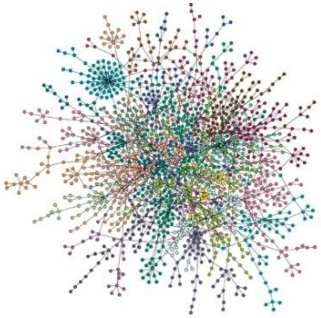
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **digraph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow A_{ij} = 1\ if\ (i, j) \in E(G)$.



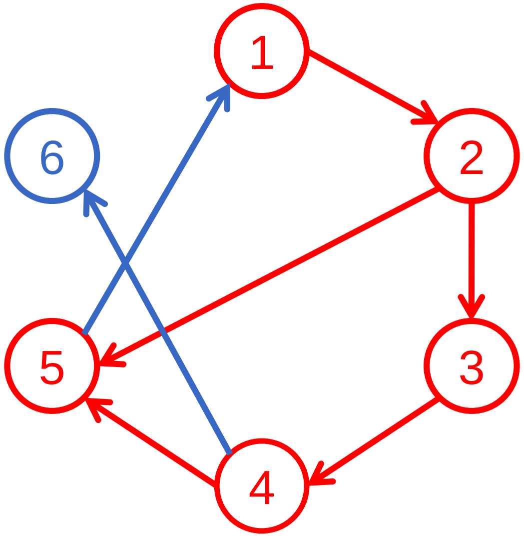
	1	2	3	4	5	6
1	0	1	0	0	0	0
2	0	0	1	0	1	0
3	0	0	0	1	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Basics…

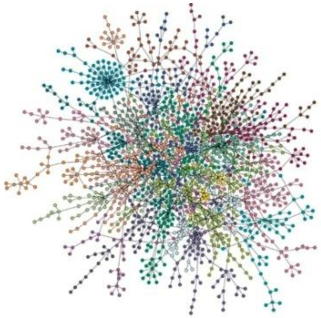


Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **digraph** G is represented by a $n \times n$ matrix, let A, where $\forall\ i, j \in V(G) \rightarrow A_{ij} = 1\ if\ (i, j) \in E(G)$.



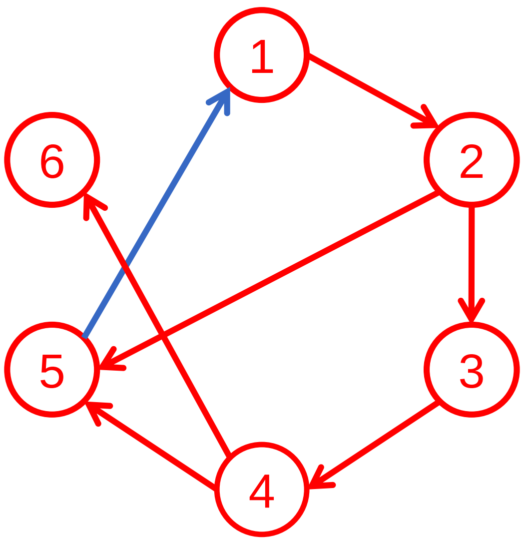
	1	2	3	4	5	6
1	0	1	0	0	0	0
2	0	0	1	0	1	0
3	0	0	0	1	0	0
4	0	0	0	0	1	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0



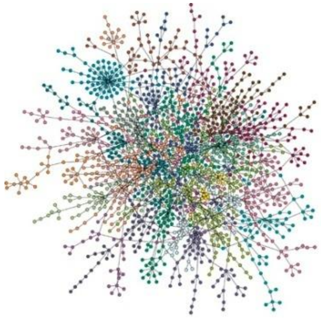
BASICS…

Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **digraph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow A_{ij} = 1\ if\ (i, j) \in E(G)$.



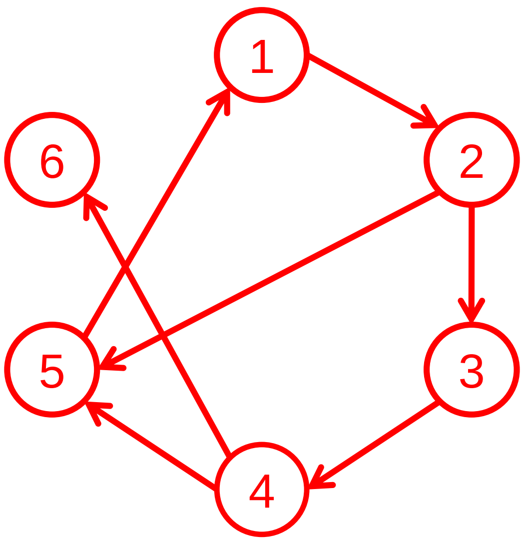
	1	2	3	4	5	6
1	0	1	0	0	0	0
2	0	0	1	0	1	0
3	0	0	0	1	0	0
4	0	0	0	0	1	1
5	0	0	0	0	0	0
6	0	0	0	0	0	0



Basics…

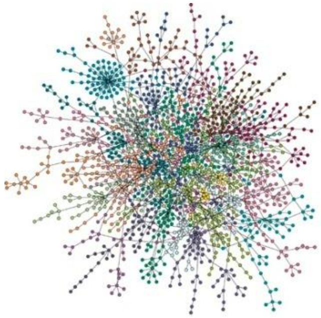
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **digraph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow A_{ij} = 1\ if\ (i, j) \in E(G)$.

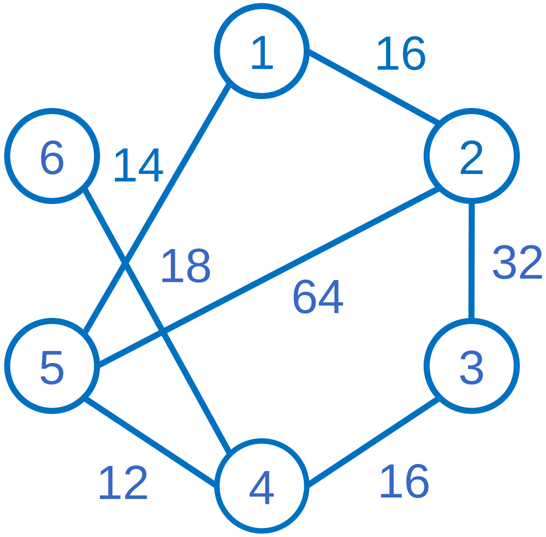


	1	2	3	4	5	6
1	0	1	0	0	0	0
2	0	0	1	0	1	0
3	0	0	0	1	0	0
4	0	0	0	0	1	1
5	1	0	0	0	0	0
6	0	0	0	0	0	0

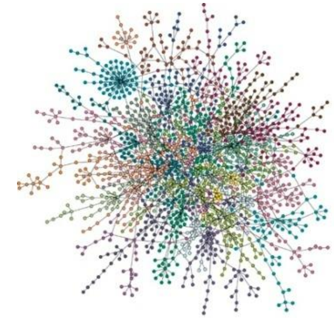
Basics…



- **Graph Representation – Adjacency Matrix**
  - Let undirected graph, G = (V, E):
    - V = vertices - $V(G)$,
    - E = edges between pairs of vertices - $E(G)$,
    - Size parameters: $n = |V|, m = |E|$.
  - A **weighted graph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow A_{ij} = w_{ij}\ if\ (i,j) \in E(G)$.



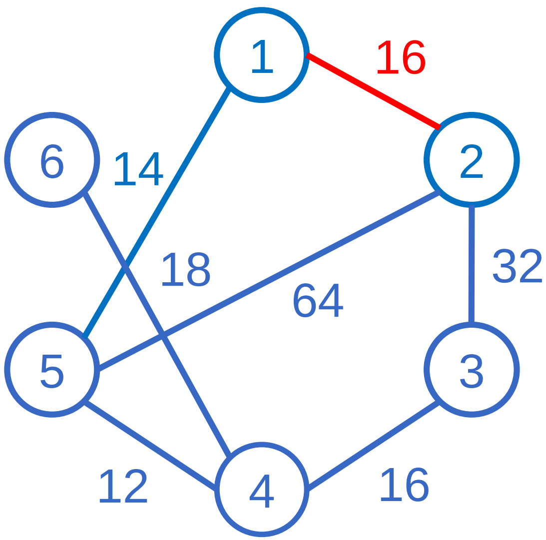
	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0



Basics…

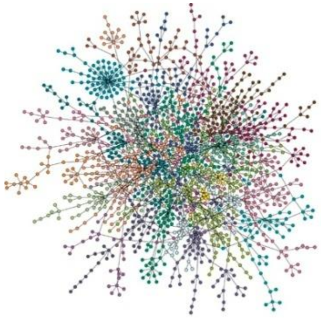
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **weighted graph** G is represented by a $n \times n$ matrix, let A, where $\forall\ i, j \in V(G) \rightarrow A_{ij} = w_{ij}\ if\ (i, j) \in E(G)$.



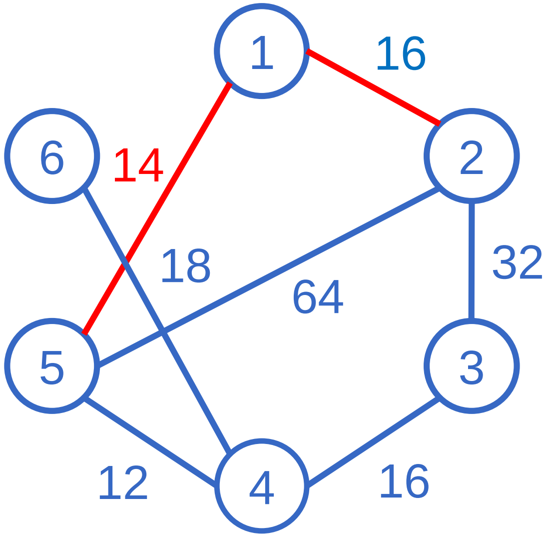
	1	2	3	4	5	6
1	0	16	0	0	0	0
2	16	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Basics…



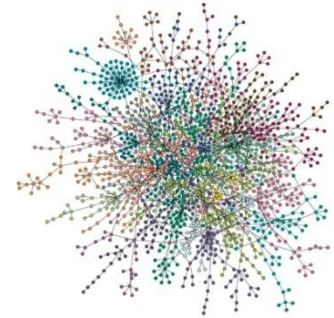
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **weighted graph** G is represented by a $n \times n$ matrix, let A, where $\forall\ i, j \in V(G) \rightarrow A_{ij} = w_{ij}\ if\ (i, j) \in E(G)$.



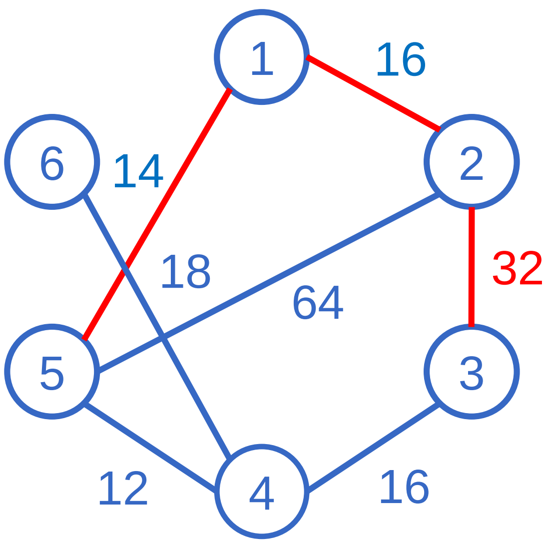
	1	2	3	4	5	6
1	0	16	0	0	14	0
2	16	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	14	0	0	0	0	0
6	0	0	0	0	0	0

Basics…



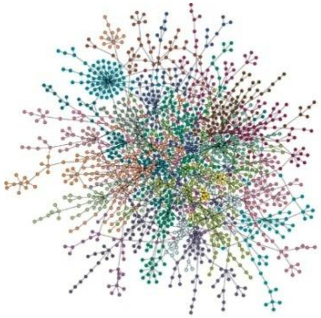
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **weighted graph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow A_{ij} = w_{ij}\ if\ (i, j) \in E(G)$.



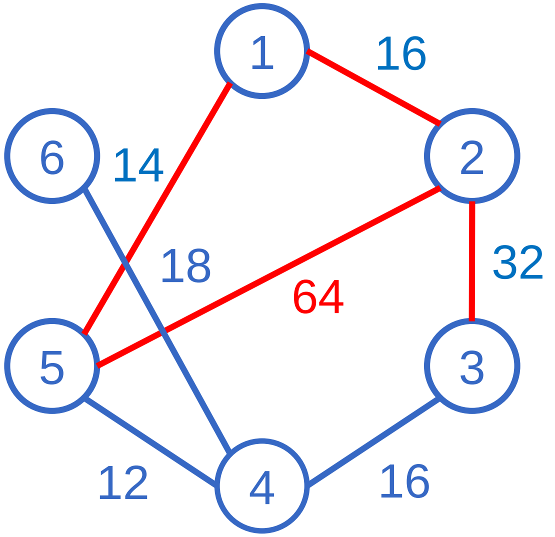
	1	2	3	4	5	6
1	0	16	0	0	14	0
2	16	0	32	0	0	0
3	0	32	0	0	0	0
4	0	0	0	0	0	0
5	14	0	0	0	0	0
6	0	0	0	0	0	0

Basics…



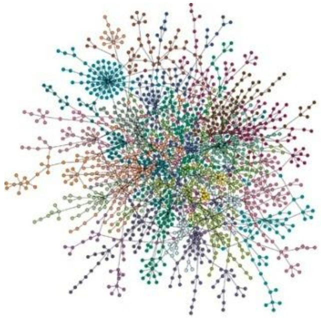
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **weighted graph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow A_{ij} = w_{ij}\ if\ (i, j) \in E(G)$.



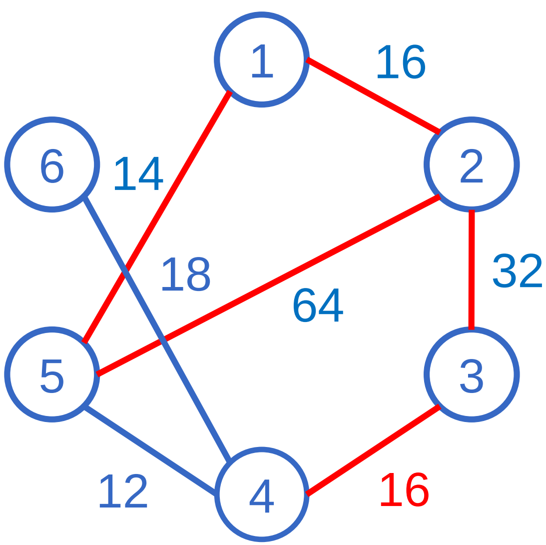
	1	2	3	4	5	6
1	0	16	0	0	14	0
2	16	0	32	0	64	0
3	0	32	0	0	0	0
4	0	0	0	0	0	0
5	14	64	0	0	0	0
6	0	0	0	0	0	0

BASICS…



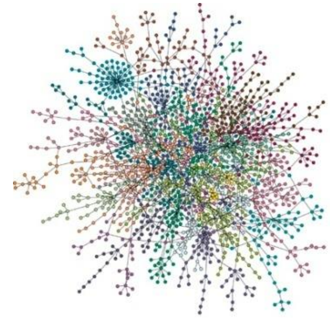
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **weighted graph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow A_{ij} = w_{ij}\ if\ (i, j) \in E(G)$.



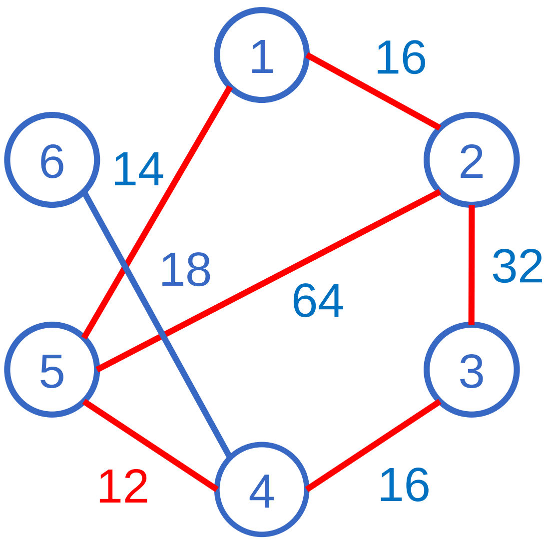
	1	2	3	4	5	6
1	0	16	0	0	14	0
2	16	0	32	0	64	0
3	0	32	0	16	0	0
4	0	0	16	0	0	0
5	14	64	0	0	0	0
6	0	0	0	0	0	0

Basics…



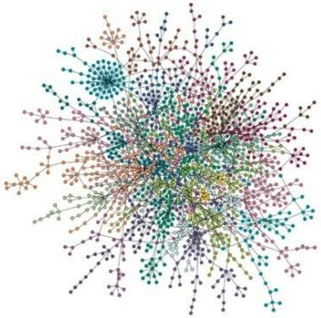
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **weighted graph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow A_{ij} = w_{ij}\ if\ (i, j) \in E(G)$.



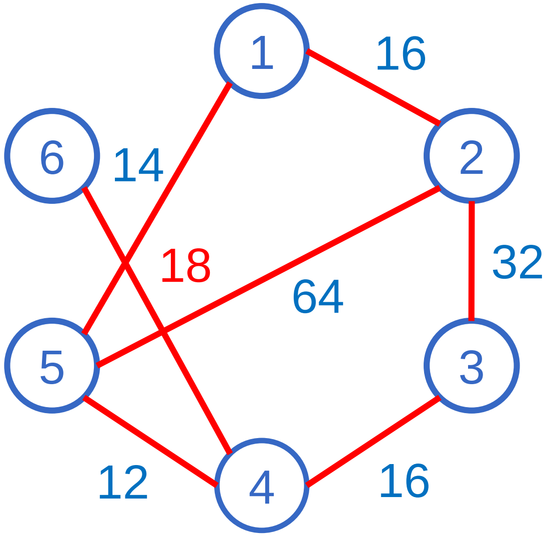
	1	2	3	4	5	6
1	0	16	0	0	14	0
2	16	0	32	0	64	0
3	0	32	0	16	0	0
4	0	0	16	0	12	0
5	14	64	0	12	0	0
6	0	0	0	0	0	0

Basics…



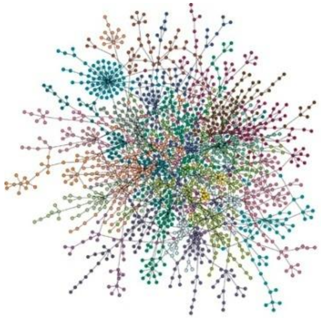
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **weighted graph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow A_{ij} = w_{ij}\ if\ (i, j) \in E(G)$.



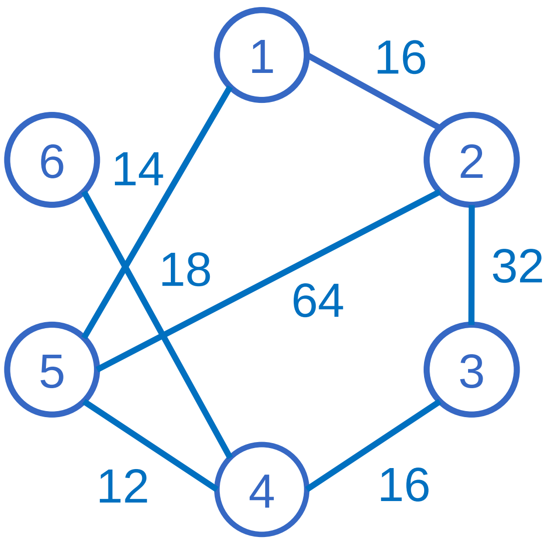
	1	2	3	4	5	6
1	0	16	0	0	14	0
2	16	0	32	0	64	0
3	0	32	0	16	0	0
4	0	0	16	0	12	18
5	14	64	0	12	0	0
6	0	0	0	18	0	0

Basics…



Graph Representation – Adjacency Matrix

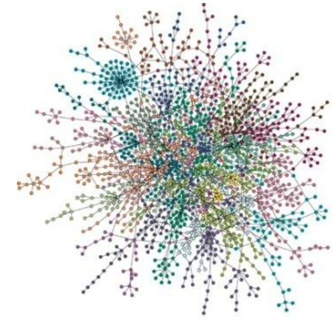
- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **weighted graph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow A_{ij} = w_{ij}\ if\ (i, j) \in E(G)$.



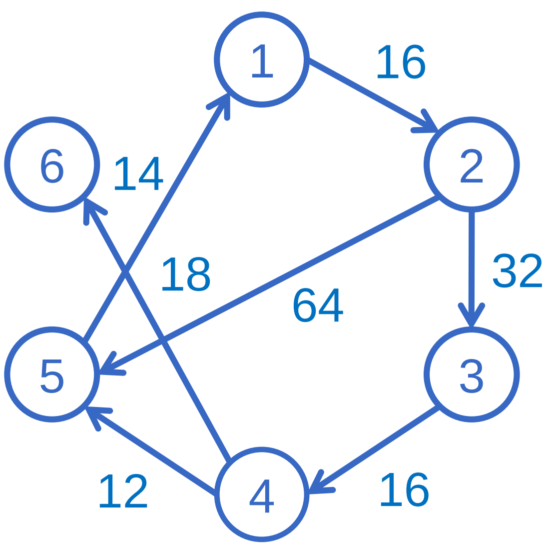
	1	2	3	4	5	6
1	0	16	0	0	14	0
2	16	0	32	0	64	0
3	0	32	0	16	0	0
4	0	0	16	0	12	18
5	14	64	0	12	0	0
6	0	0	0	18	0	0

Basics…

Graph Representation – Adjacency Matrix

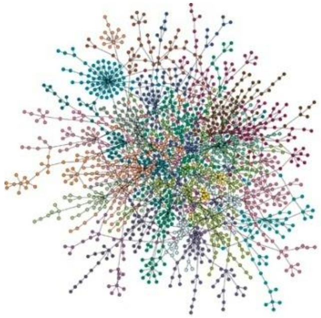


- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **weighted digraph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow A_{ij} = 1\; if\; (i, j) \in E(G)$.



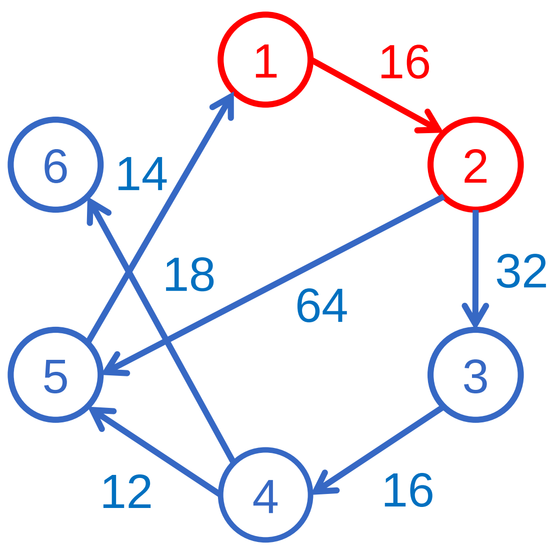
	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Basics…



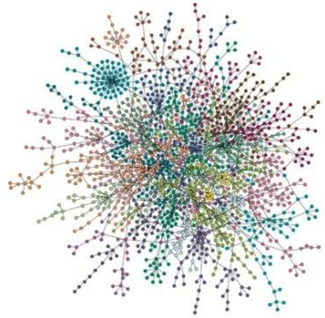
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **weighted digraph** G is represented by a $n \times n$ matrix, let A, where $\forall\ i,j \in V(G) \rightarrow A_{ij} = 1\ if\ (i,j) \in E(G)$.



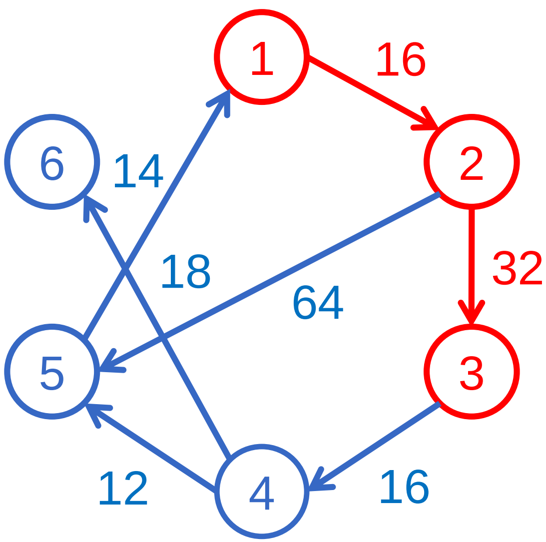
	1	2	3	4	5	6
1	0	16	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Basics…



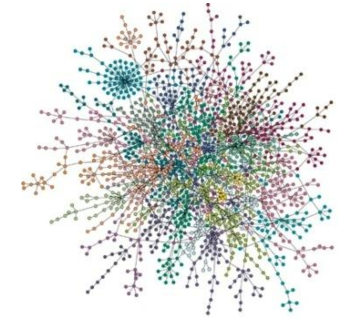
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **weighted digraph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i,j \in V(G) \rightarrow A_{ij} = 1 \; if \; (i,j) \in E(G)$.



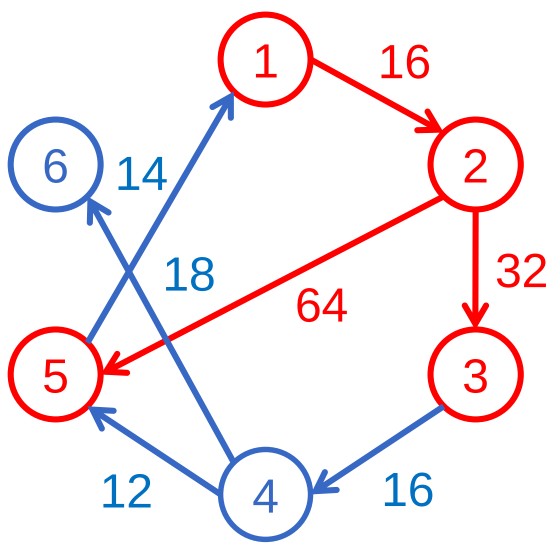
	1	2	3	4	5	6
1	0	16	0	0	0	0
2	0	0	32	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

BASICS…



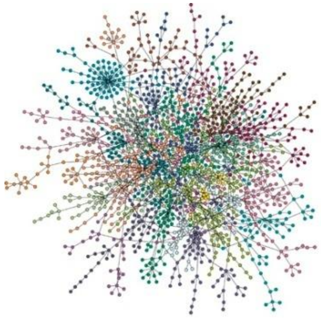
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **weighted digraph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow A_{ij} = 1\ if\ (i, j) \in E(G)$.

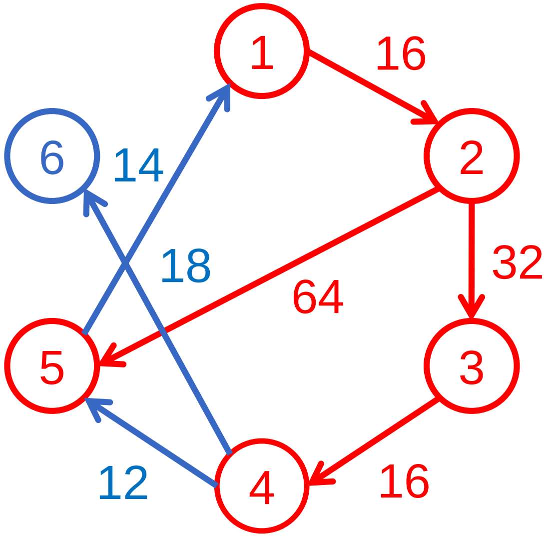


	1	2	3	4	5	6
1	0	16	0	0	0	0
2	0	0	32	0	64	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Basics…

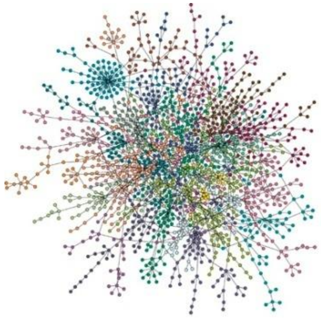


- **Graph Representation – Adjacency Matrix**
  - Let undirected graph, G = (V, E):
    - V = vertices - $V(G)$,
    - E = edges between pairs of vertices - $E(G)$,
    - Size parameters: $n = |V|, m = |E|$.
  - A **weighted digraph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i,j \in V(G) \rightarrow A_{ij} = 1\ if\ (i,j) \in E(G)$.



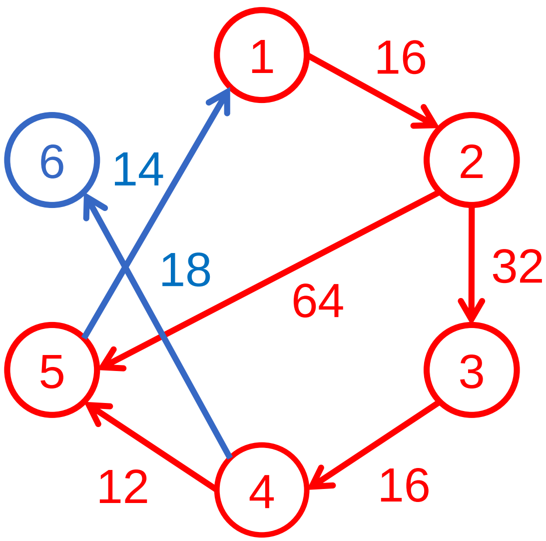
	1	2	3	4	5	6
1	0	16	0	0	0	0
2	0	0	32	0	64	0
3	0	0	0	16	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Basics…



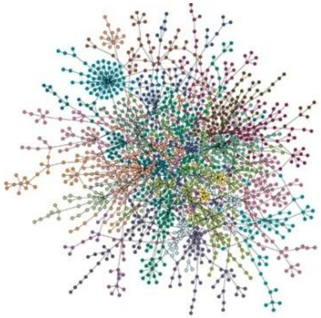
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **weighted digraph** G is represented by a $n \times n$ matrix, let A, where $\forall\ i, j\ \in\ V(G)\ \rightarrow A_{ij} = 1\ if\ (i, j)\ \in\ E(G)$.



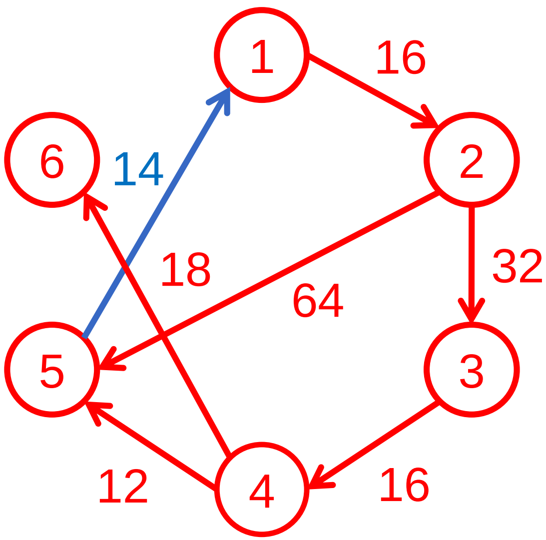
	1	2	3	4	5	6
1	0	16	0	0	0	0
2	0	0	32	0	64	0
3	0	0	0	16	0	0
4	0	0	0	0	12	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Basics…



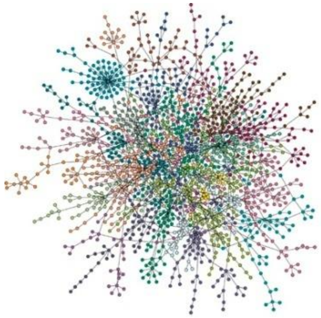
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **weighted digraph** G is represented by a $n \times n$ matrix, let A, where $\forall\ i,j \in V(G) \rightarrow A_{ij} = 1\ if\ (i,j) \in E(G)$.



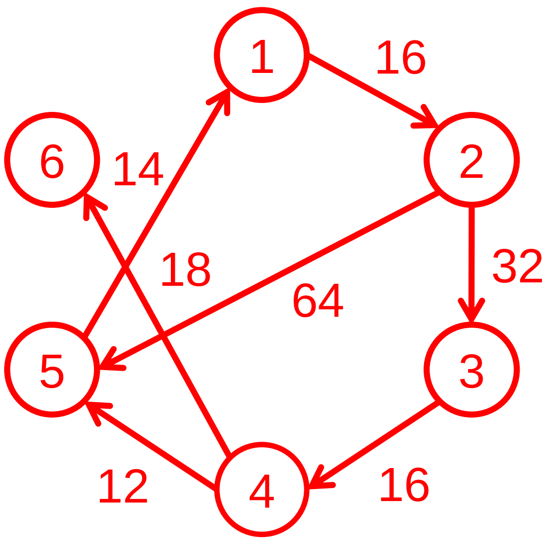
	1	2	3	4	5	6
1	0	16	0	0	0	0
2	0	0	32	0	64	0
3	0	0	0	16	0	0
4	0	0	0	0	12	18
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Basics…



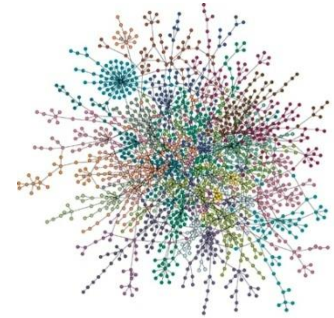
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **weighted digraph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i, j \in V(G) \rightarrow A_{ij} = 1\; if\; (i, j) \in E(G)$.



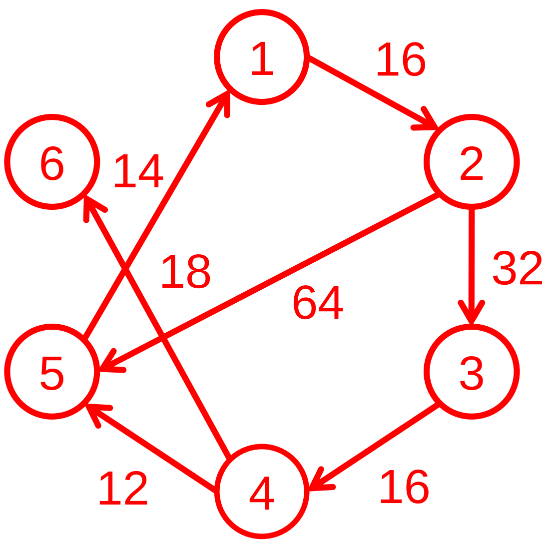
	1	2	3	4	5	6
1	0	16	0	0	0	0
2	0	0	32	0	64	0
3	0	0	0	16	0	0
4	0	0	0	0	12	18
5	14	0	0	0	0	0
6	0	0	0	0	0	0

Basics…



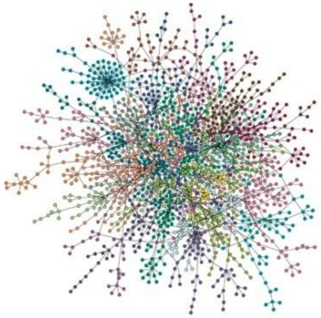
Graph Representation – Adjacency Matrix

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- A **weighted digraph** G is represented by a $n \times n$ matrix, let A, where $\forall\, i,j \in V(G) \rightarrow A_{ij} = 1\ if\ (i,j) \in E(G)$.



	1	2	3	4	5	6
1	0	16	0	0	0	0
2	0	0	32	0	64	0
3	0	0	0	16	0	0
4	0	0	0	0	12	18
5	14	0	0	0	0	0
6	0	0	0	0	0	0

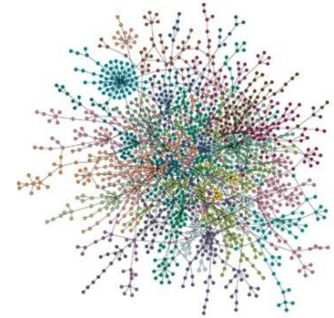
BASICS…



- **Graph Representation – Adjacency List**
  - Let undirected graph, G = (V, E):
    - V = vertices - $V(G)$,
    - E = edges between pairs of vertices - $E(G)$,
    - Size parameters: $n = |V|, m = |E|$.
  - The Adjacency List L of a graph G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

$$\forall\, v_i, vj \in V(G) \rightarrow v_j \in l_i \; iff \; (v_i, vj) \in E(G), \text{where } i = |V|.$$

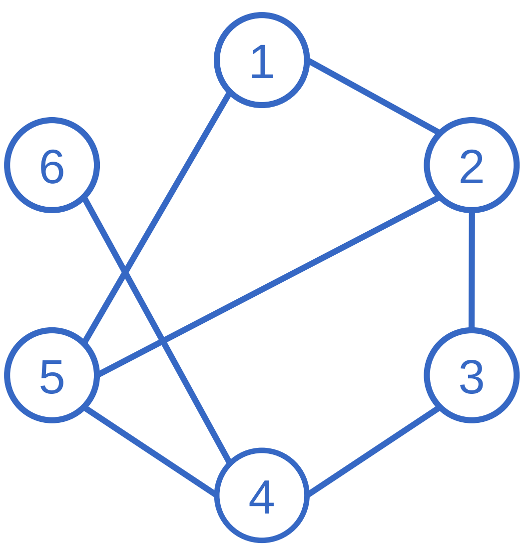
BASICS…

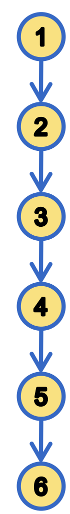


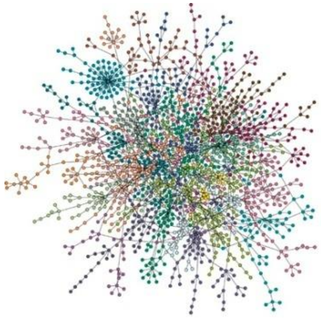
Graph Representation – Adjacency List

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- The Adjacency List L of a graph G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

$$\forall\, v_i, vj \in V(G) \rightarrow v_j \in l_i \; iff \; (v_i, vj) \in E(G), \text{ where } i = |V|.$$



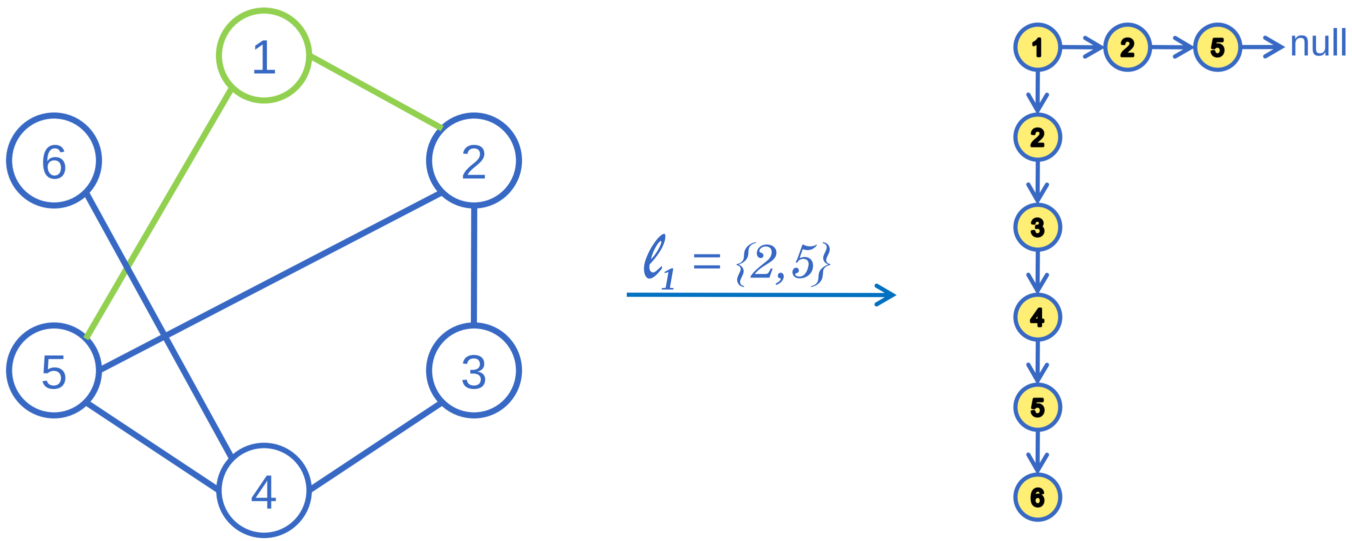




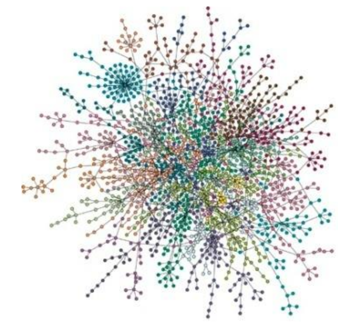
Graph Representation – Adjacency List

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- The Adjacency List L of a graph G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

$$\forall\, v_i, vj \in V(G) \rightarrow v_j \in l_i \; iff \; (v_i, vj) \in E(G), \text{where } i = |V|.$$



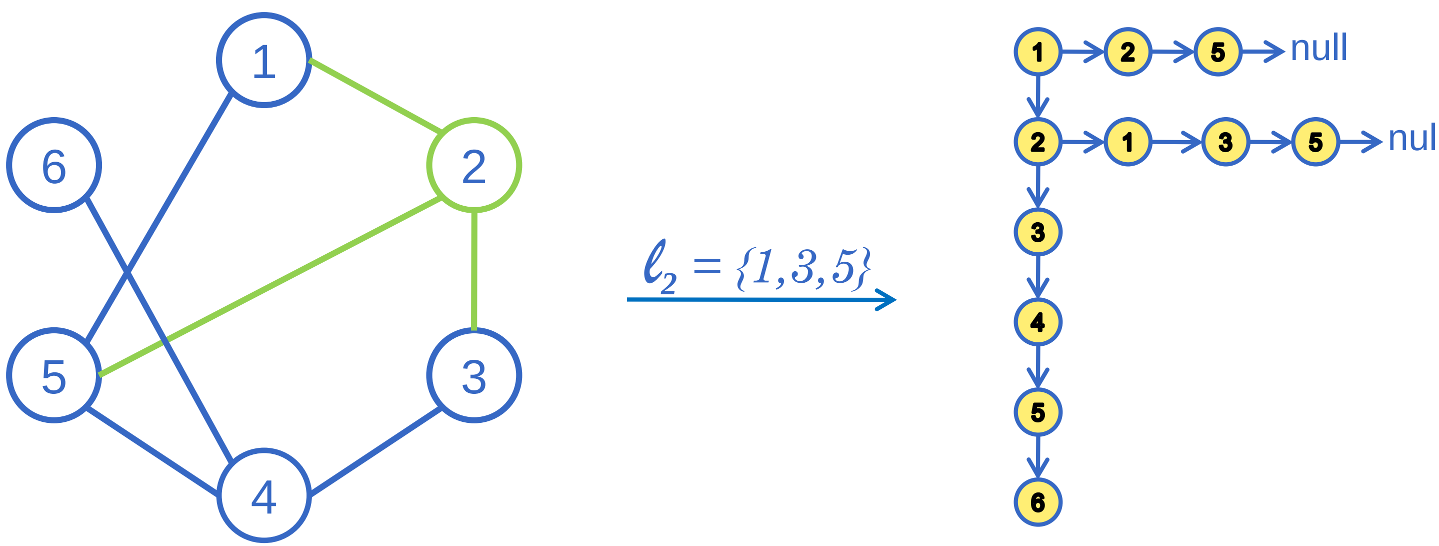
BASICS…

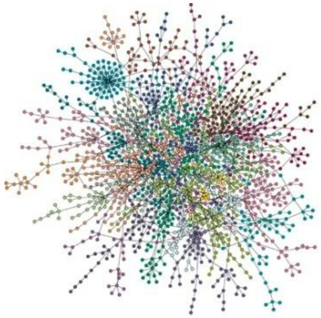


Graph Representation – Adjacency List

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- The Adjacency List L of a graph G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

$$\forall\, v_i, vj \in V(G) \rightarrow v_j \in l_i \; iff \; (v_i, vj) \in E(G), \text{ where } i = |V|.$$

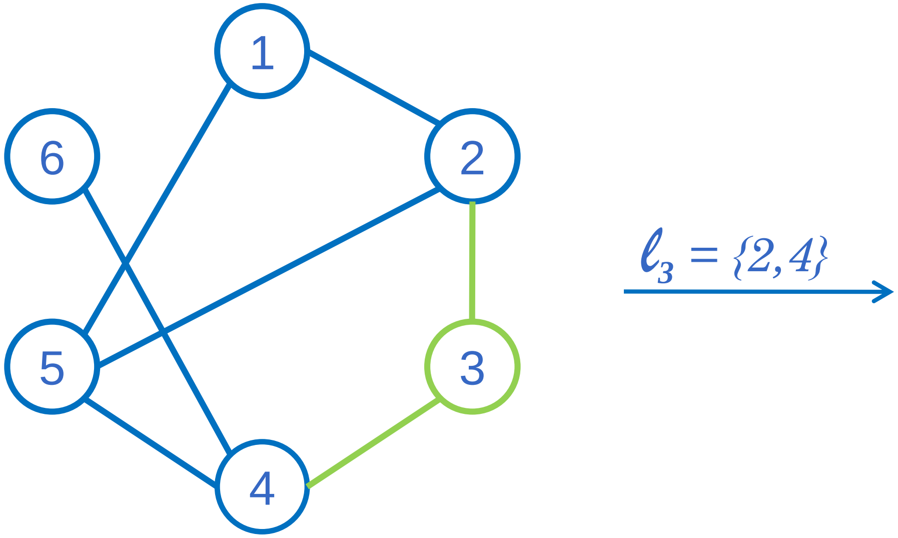


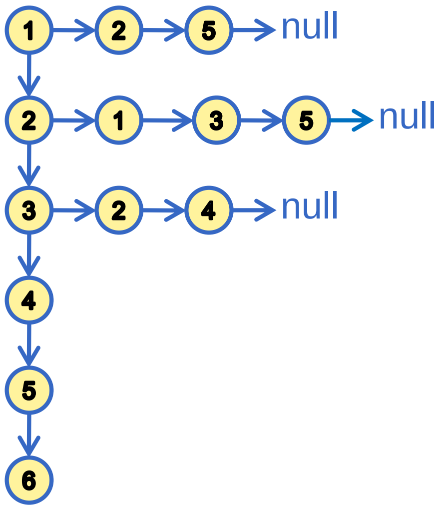


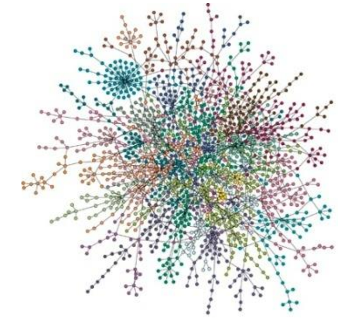
Graph Representation – Adjacency List

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- The Adjacency List L of a graph G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

$$\forall\, v_i, vj \in V(G) \rightarrow v_j \in l_i \; iff \; (v_i, vj) \in E(G), \text{ where } i = |V|.$$



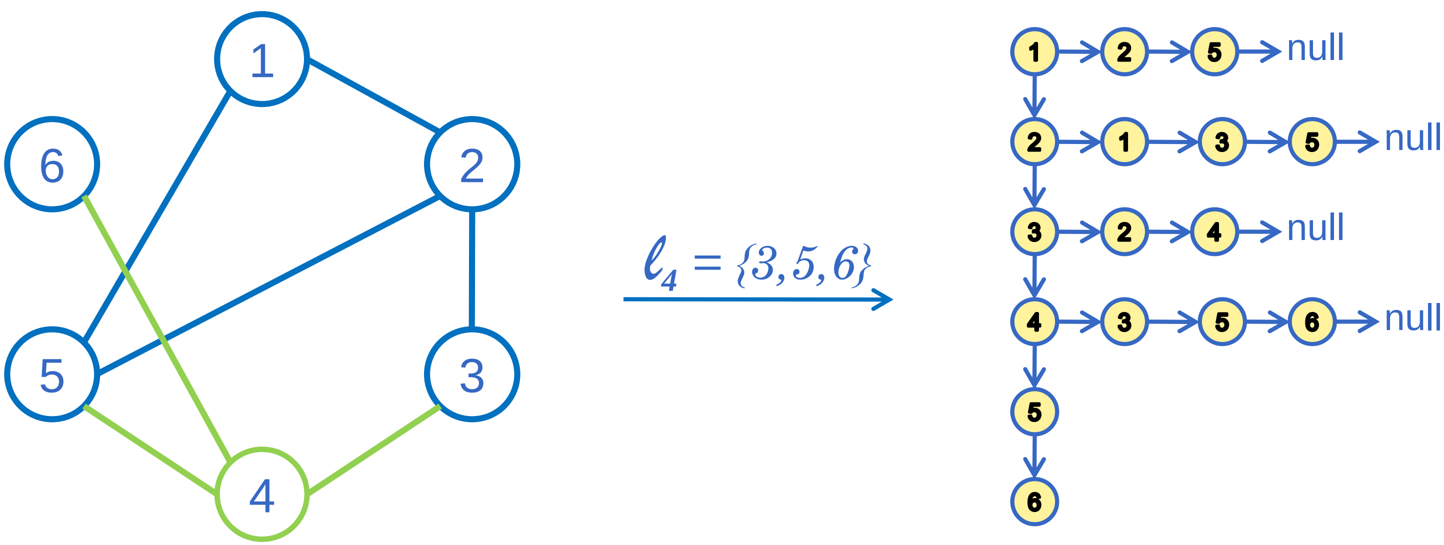




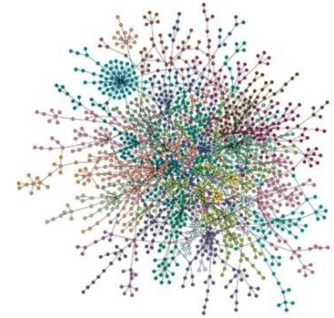
Graph Representation – Adjacency List

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- The Adjacency List L of a graph G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

$$\forall\, v_i, vj \in V(G) \rightarrow v_j \in l_i \; iff \; (v_i, vj) \in E(G), \text{ where } i = |V|.$$



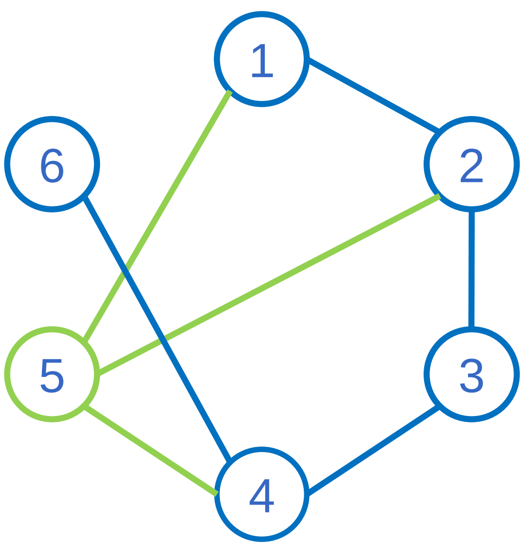
Basics…



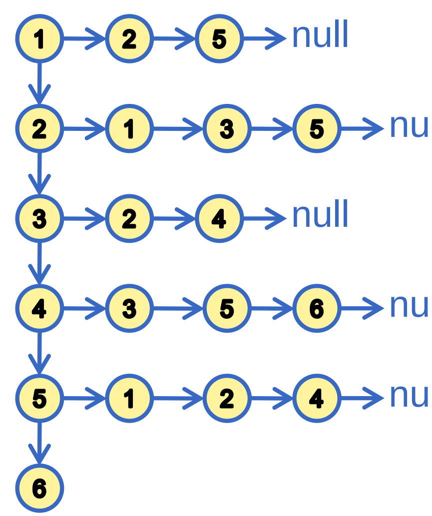
Graph Representation – Adjacency List

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- The Adjacency List L of a graph G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

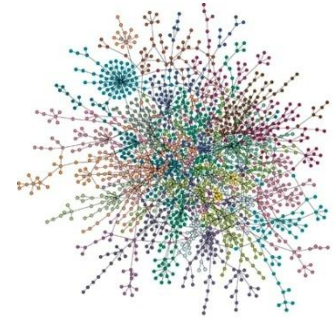
$$\forall\, v_i, vj \in V(G) \rightarrow v_j \in l_i \; iff \; (v_i, vj) \in E(G), \text{where } i = |V|.$$



$\ell_5 = \{1,2,4\}$ →



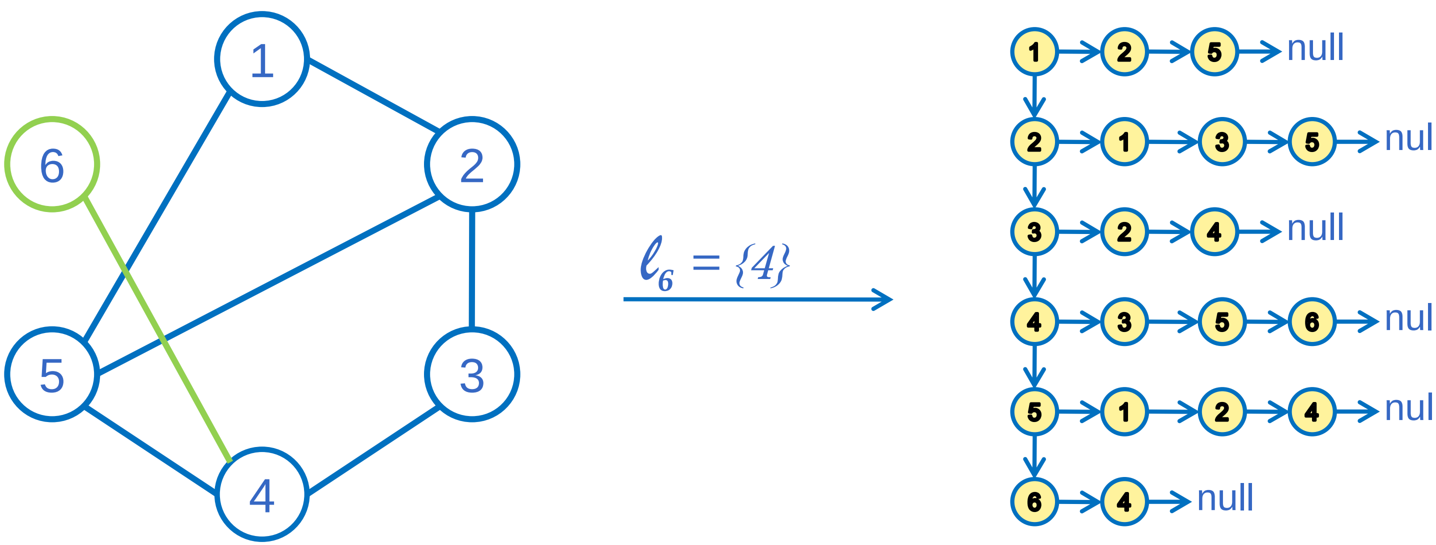
Basics…



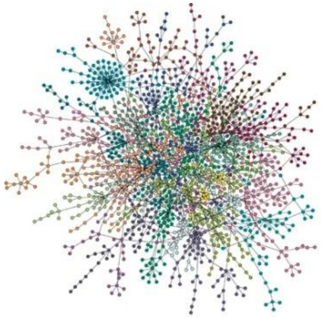
Graph Representation – Adjacency List

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- The Adjacency List L of a graph G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

$$\forall\, v_i, vj \in V(G) \rightarrow v_j \in l_i \; iff \; (v_i, vj) \in E(G), \text{where } i = |V|.$$

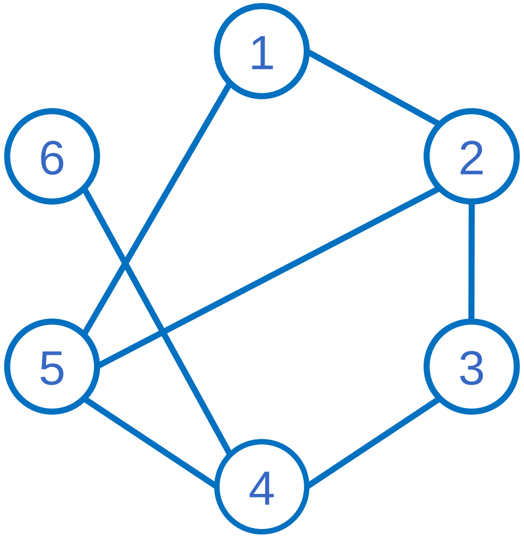


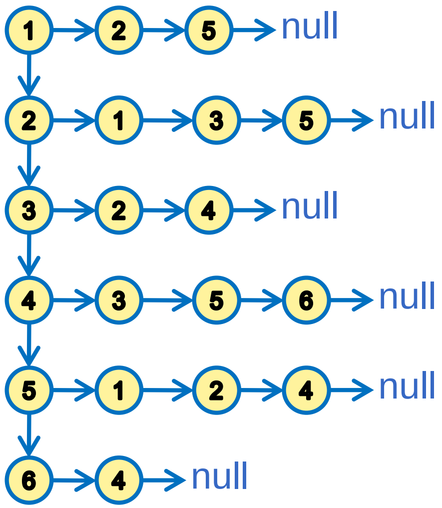
BASICS…



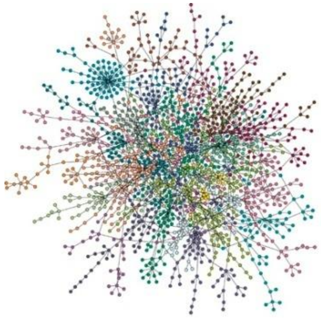
- **Graph Representation – Adjacency List**
  - Let undirected graph, G = (V, E):
    - V = vertices - $V(G)$,
    - E = edges between pairs of vertices - $E(G)$,
    - Size parameters: $n = |V|, m = |E|$.
  - The Adjacency List L of a graph G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

$$\forall\, v_i, vj \in V(G) \rightarrow v_j \in l_i \; iff \; (v_i, vj) \in E(G), \text{ where } i = |V|.$$





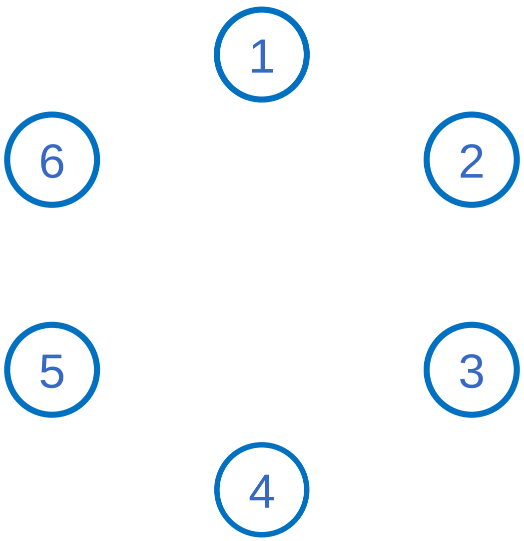
Basics…

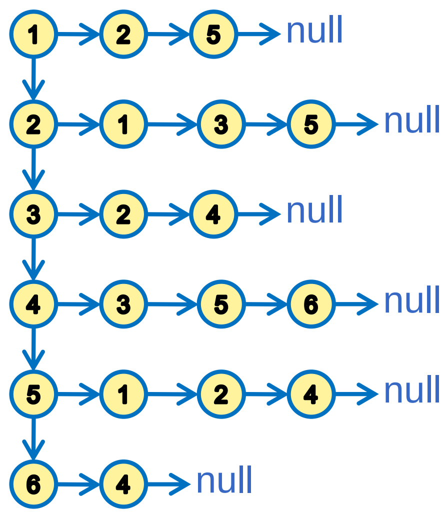


Graph Representation – Adjacency List

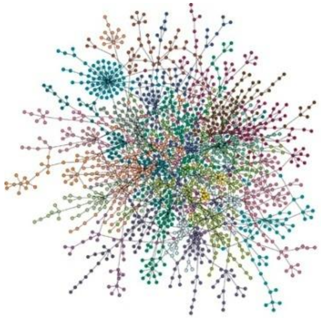
- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- The Adjacency List L of a graph G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

$$\forall v_i, v_j \in V(G) \rightarrow v_j \in l_i \; iff \; (v_i, v_j) \in E(G), \text{ where } i = |V|.$$





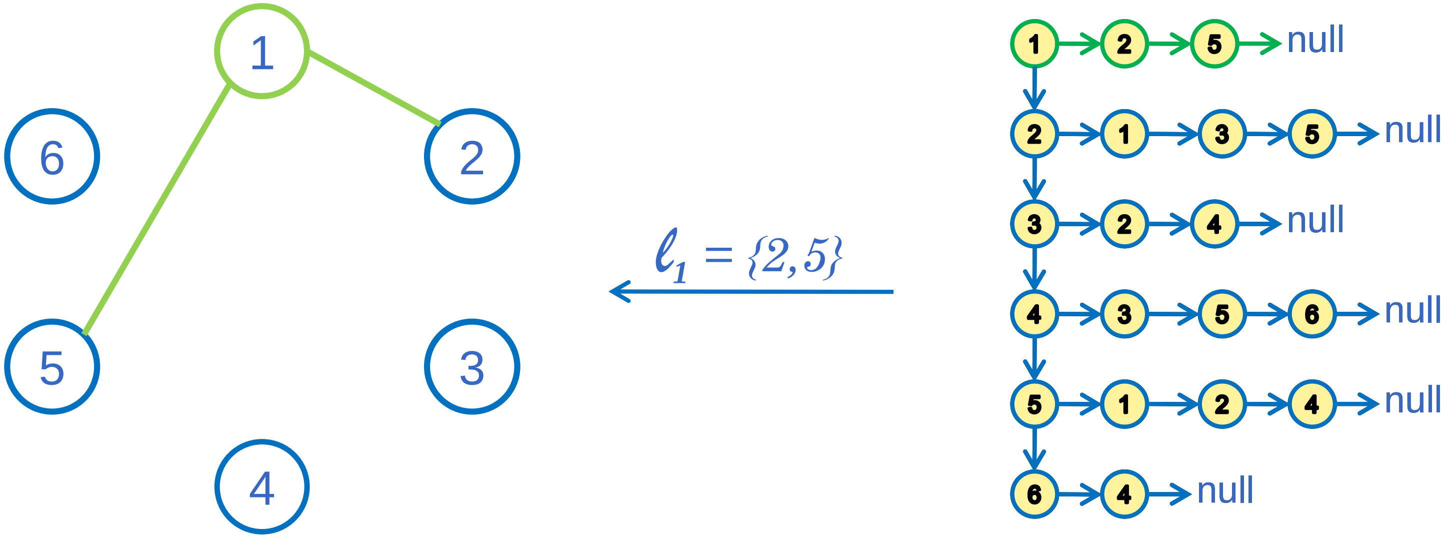
Basics…



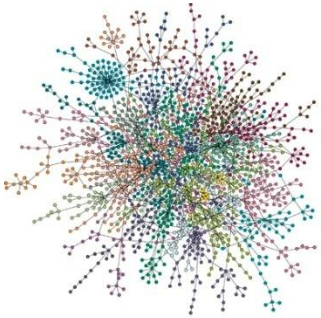
Graph Representation – Adjacency List

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- The Adjacency List L of a graph G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

$$\forall v_i, v_j \in V(G) \rightarrow v_j \in l_i \; iff \; (v_i, v_j) \in E(G), \text{ where } i = |V|.$$



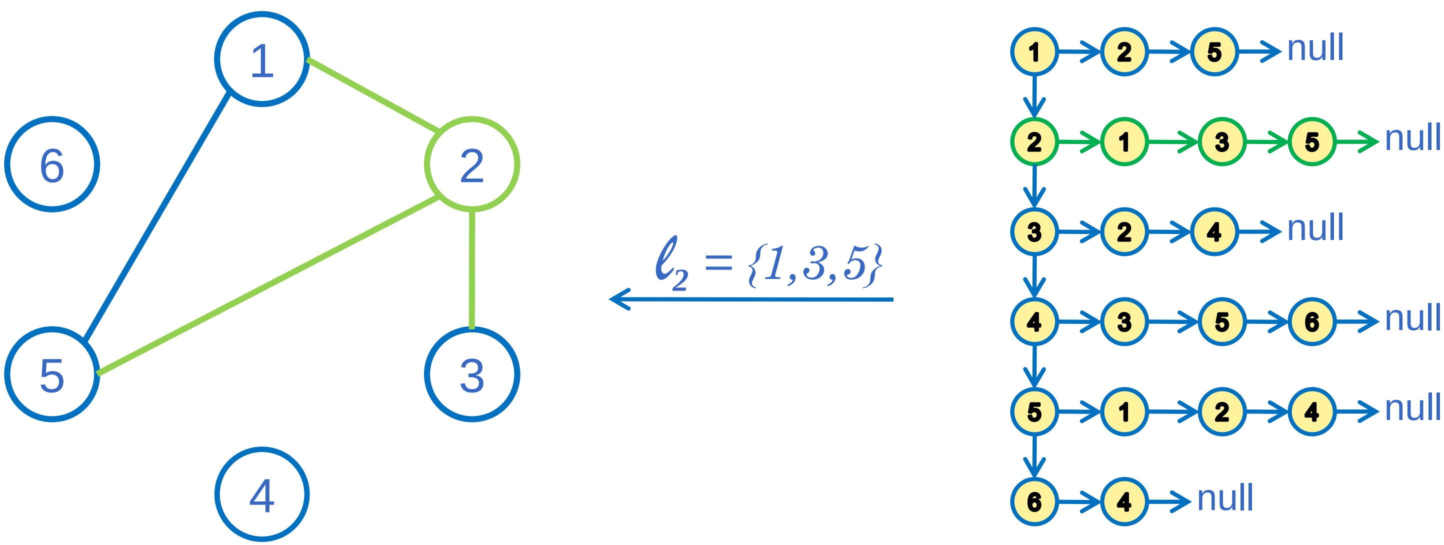
Basics…

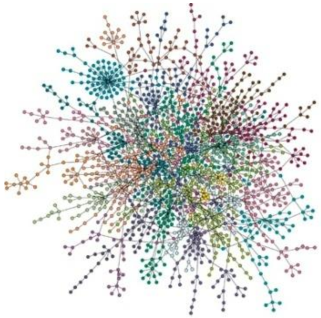


Graph Representation – Adjacency List

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- The Adjacency List L of a graph G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

$$\forall v_i, v_j \in V(G) \rightarrow v_j \in l_i \; iff \; (v_i, v_j) \in E(G), \text{where } i = |V|.$$



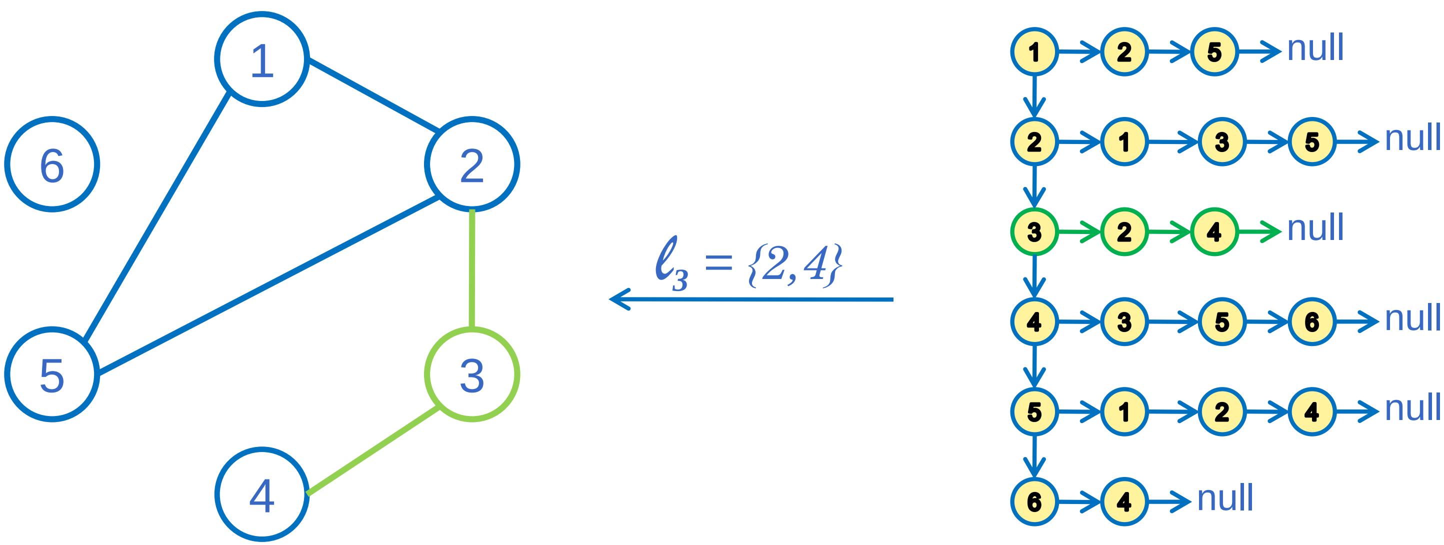


Basics…

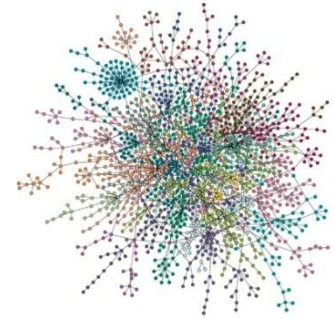
Graph Representation – Adjacency List

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- The Adjacency List L of a graph G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

$$\forall v_i, v_j \in V(G) \rightarrow v_j \in l_i \; iff \; (v_i, v_j) \in E(G), \text{ where } i = |V|.$$



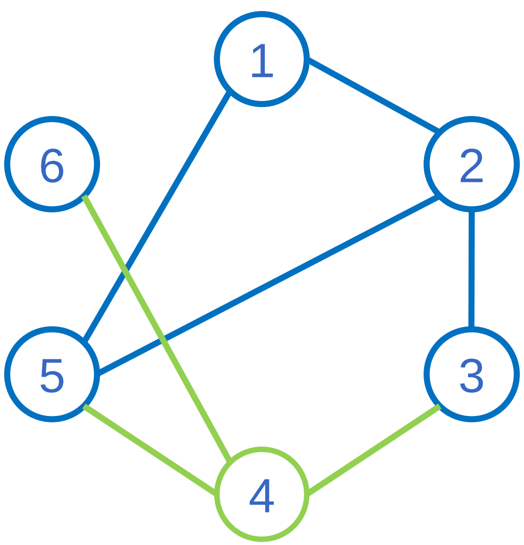
Basics…



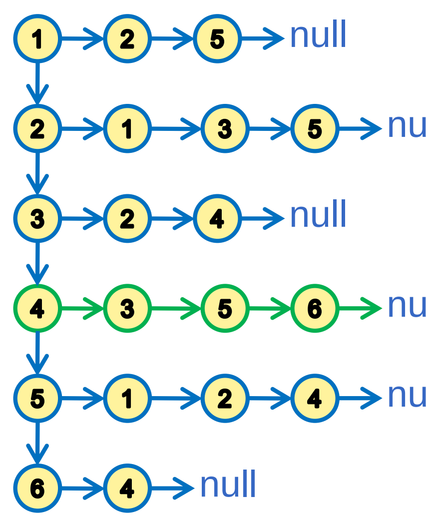
Graph Representation – Adjacency List

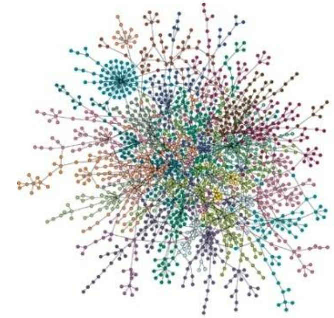
- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- The Adjacency List L of a graph G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

$$\forall v_i, v_j \in V(G) \rightarrow v_j \in l_i \text{ iff } (v_i, v_j) \in E(G), \text{where } i = |V|.$$



$\ell_4 = \{3,5,6\}$

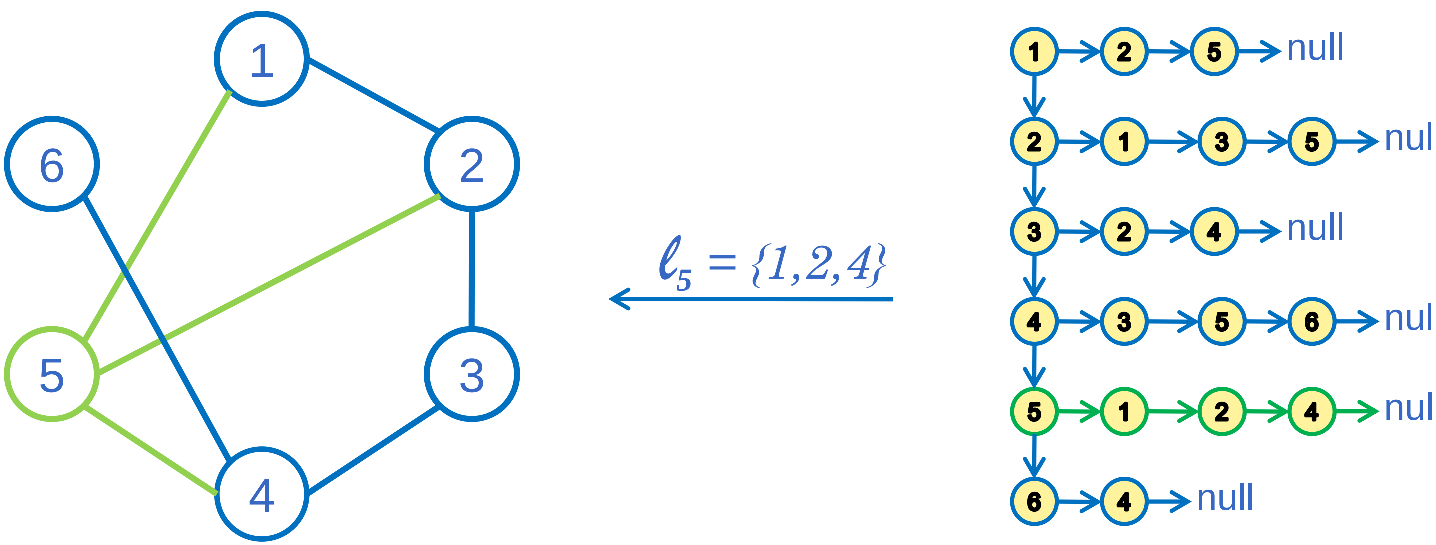




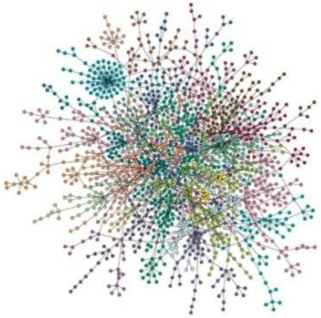
Graph Representation – Adjacency List

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- The Adjacency List L of a graph G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

$$\forall v_i, v_j \in V(G) \rightarrow v_j \in l_i \; iff \; (v_i, v_j) \in E(G), \text{where } i = |V|.$$



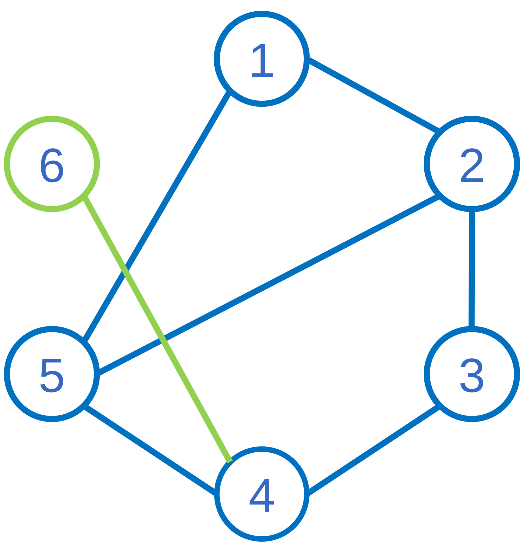
Basics…



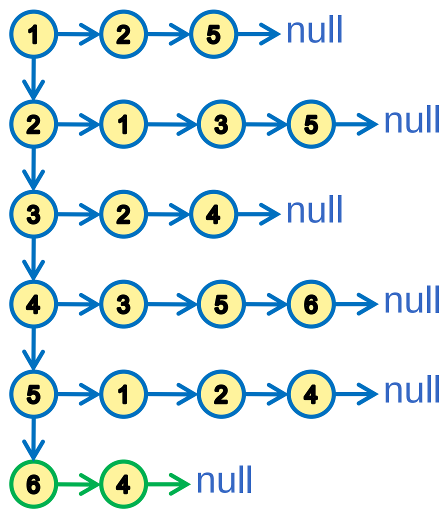
Graph Representation – Adjacency List

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- The Adjacency List L of a graph G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

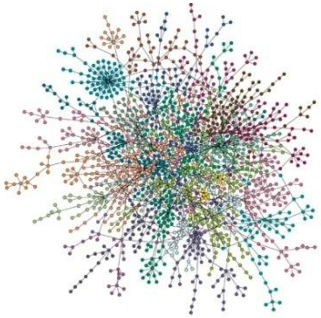
$$\forall v_i, v_j \in V(G) \rightarrow v_j \in l_i \; iff \; (v_i, v_j) \in E(G), \text{where } i = |V|.$$



$\ell_6 = \{4\}$



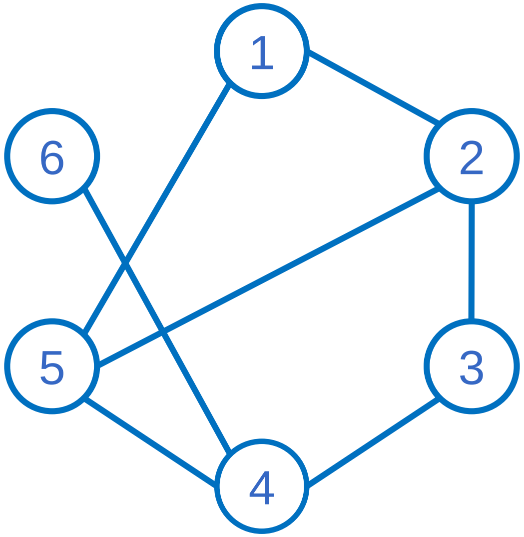
Basics…

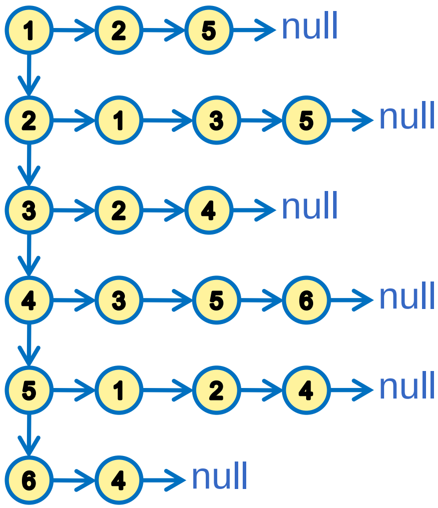


Graph Representation – Adjacency List

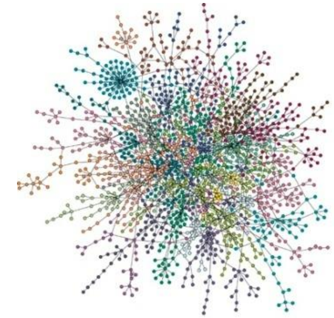
- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- The Adjacency List L of a **graph** G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

$$\forall v_i, v_j \in V(G) \rightarrow v_j \in l_i \; iff \; (v_i, v_j) \in E(G), \text{where } i = |V|.$$





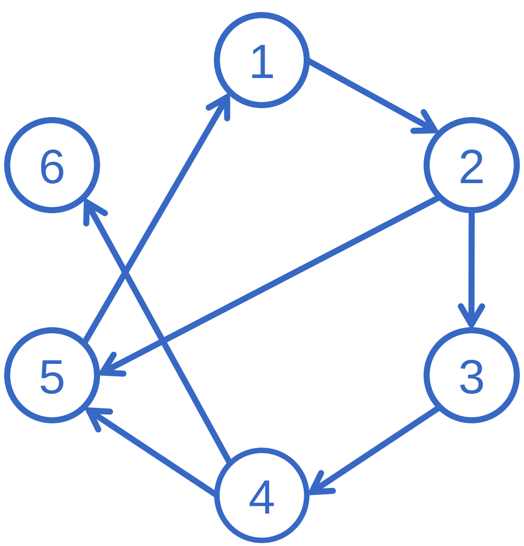
Basics…

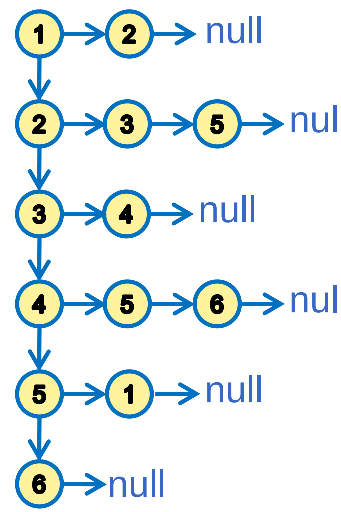


Graph Representation – Adjacency List

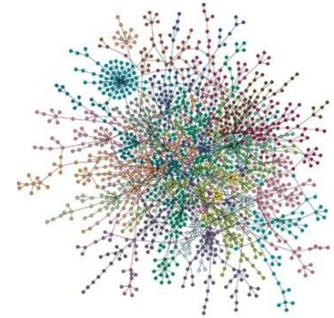
- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- The Adjacency List L of a **digraph** G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

$$\forall v_i, v_j \in V(G) \rightarrow v_j \in l_i \; iff \; (v_i, v_j) \in E(G), \text{ where } i = |V|.$$





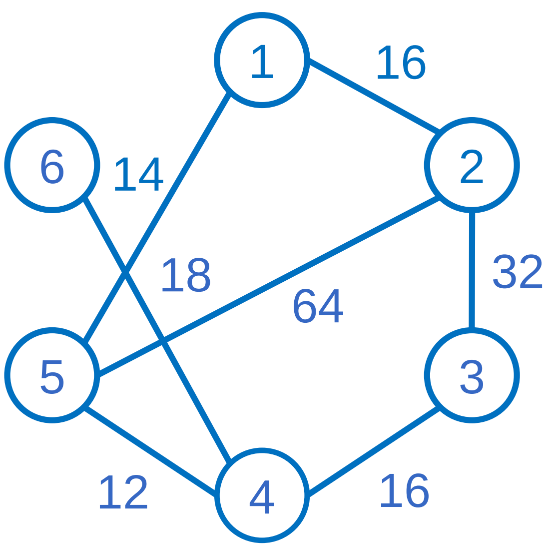
BASICS…

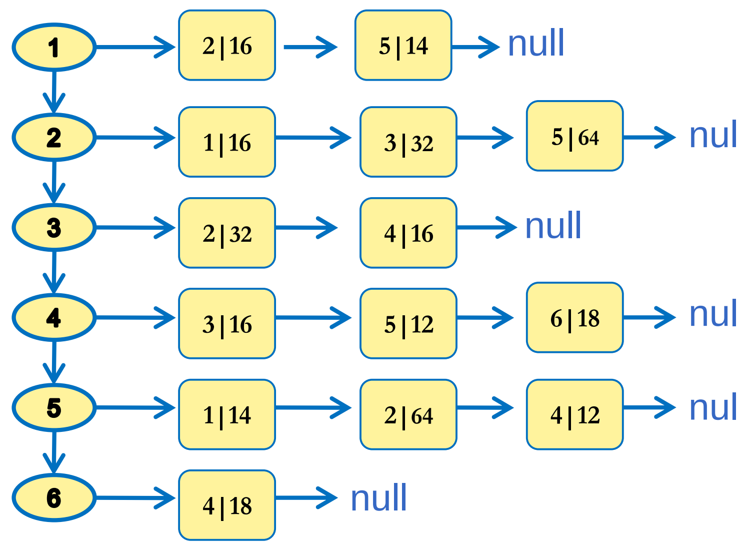


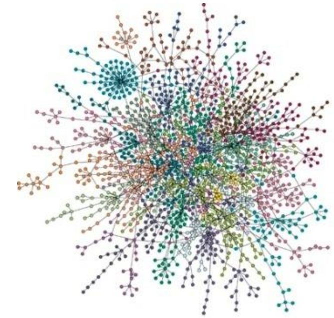
Graph Representation – Adjacency List

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- The Adjacency List L of a **weighted graph** G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

$$\forall v_i, v_j \in V(G) \rightarrow v_j \in l_i \; iff \; (v_i, v_j) \in E(G), \text{where } i = |V|.$$







Graph Representation – Adjacency List

- Let undirected graph, G = (V, E):
  - V = vertices - $V(G)$,
  - E = edges between pairs of vertices - $E(G)$,
  - Size parameters: $n = |V|, m = |E|$.
- The Adjacency List L of a **weighted digraph** G is composed by a set of lists l_i including for each vertex of the graph its adjacent vertices, as follows:

$$\forall\, v_i, v_j \in V(G) \rightarrow v_j \in l_i \;\; iff \;\; (v_i, v_j) \in E(G), \text{ where } i = |V|.$$