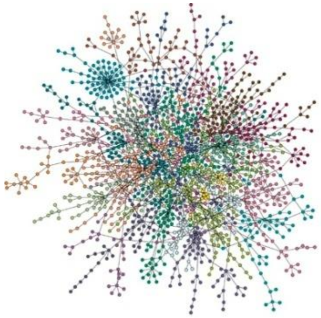
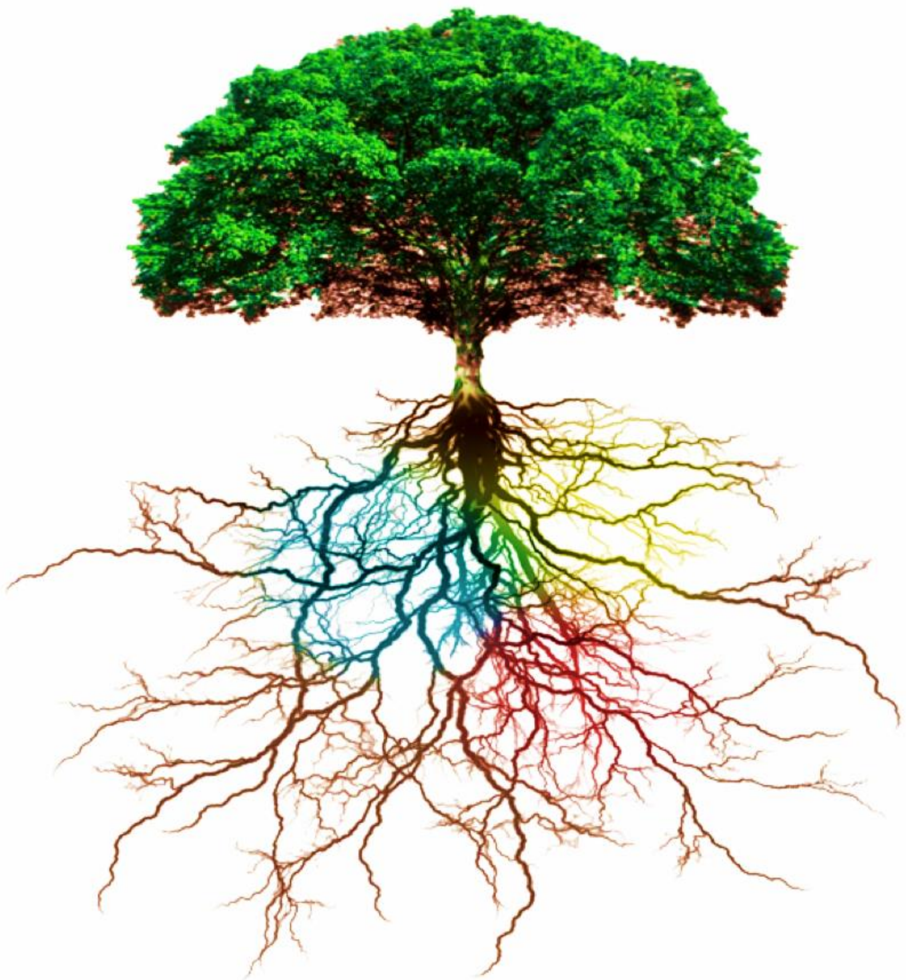
Graph Theory

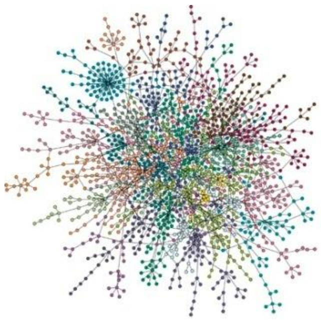
Trees

Trees





TREES

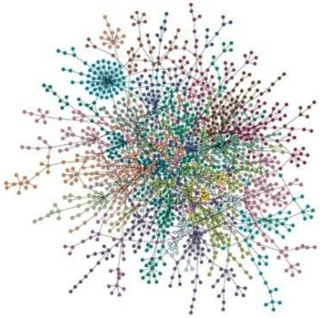


- **Basics**
  - Trees are connected acyclic graphs
  - **Theorem 1**:

    A graph G is a tree *iff* any two vertices are connected with a unique path.
  - **Theorem 2**:

    In a tree T it holds that $m = n - 1$

Trees



- **Basics**
  - Trees are connected acyclic graphs
  - **<u>Theorem 1</u>**:

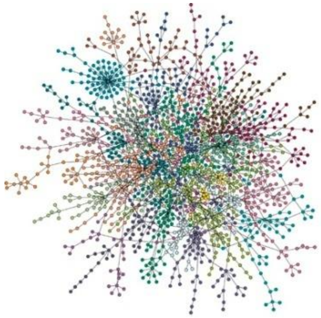
    A graph G is a tree *iff* any two vertices are connected with a unique path.
  - **<u>Theorem 2</u>**:

    In a tree T it holds that $m = n - 1$

    - For $n = 1 \rightarrow m = 0$ (it holds for n=1)
    - Assumption: Let that it holds for all the tree with $k \leq n$ vertices, $k \geq 2$
    - Let T a tree with n vertices and m edges, and $e \in T$
    - Delete e from $T \Rightarrow T - e$ results to two trees $\Rightarrow T_1(n_1, m_1)$ *and* $T_2(n_2, m_2)$
      1. from the assumption it holds that $m_1 = n_1 - 1$ *and* $m_2 = n_2 - 1$
      2. $n = n_1 + n_2$ and $m = m_1 + m_2 + 1$

    (1),(2) $\Rightarrow m = (n_1 - 1) + (n_2 - 1) + 1 = n_1 + n_2 - 1 = n - 1$

TREES



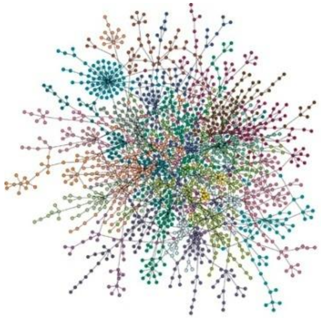
- **Basics**
  - Trees are connected acyclic graphs
  - **Theorem 1**:

    A graph G is a tree *iff* any two vertices are connected with a unique path.
  - **Theorem 2**:

    In a tree T it holds that $m = n - 1$
  - **Corollary 1**:

    In each tree there exist at least two vertices with degree 1

Trees



- **Basics**
  - Trees are connected acyclic graphs
  - **Theorem 1**:

    A graph G is a tree *iff* any two vertices are connected with a unique path.
  - **Theorem 2**:

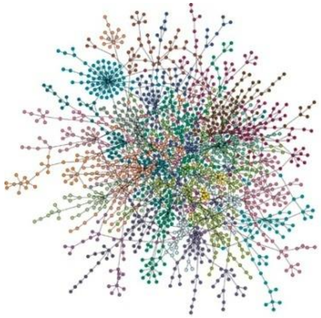
    In a tree T it holds that $m = n - 1$
  - **Corollary 1**:

    In each tree there exist at least two vertices with degree 1

    - It holds that $2m = 2(n-1)$
    - Let $d_1, d_2, \dots, d_n$ and $d_1 = 1$ and $d_i \geq 2$
    - Then $\sum_{i=1}^{n} d_i \geq 1 + 2(n-1) = 2n - 1$
    - it holds $\sum_{i=1}^{n} d_i = 2m = 2(n-1) = 2n - 2$

    $\Bigg\}$ **Contradiction!**

Trees



Basics

- Trees are connected acyclic graphs

- **Theorem 1**:

  A graph G is a tree *iff* any two vertices are connected with a unique path.

- **Theorem 2**:

  In a tree T it holds that $m = n - 1$

- **Corollary 1**:

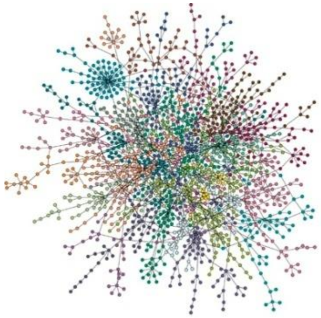
  In each tree there exist at least two vertices with degree 1.

- **Corollary 2 (necessary but not sufficient condition)**:

  A non increasing sequence of integers, let S: $d_1, d_2, \dots, d_n$ belongs to a tree only if every d_i is a positive integer and it holds that

$$\sum_{i=1} d_i = 2(n-1)$$

TREES

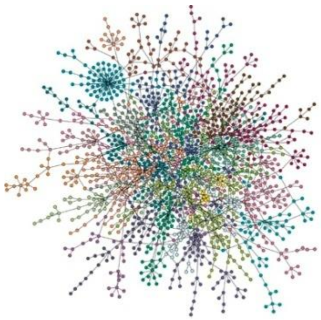


- **Basics**
  - **Theorem 3**:

    An acyclic graph G with n vertices and $n-1$ edges is connected

    - Let that there exist a graph G with n vertices and $n-1$ edges which is not connected.
    - In that case, it is composed by two or more components.
    - Let that it is composed by two components G_1 and G_2.
    - We assume an edge let (v, w): $v \in G_1$ and $w \in G_2$. So, no cycle is constructed, since before there was no other path between v and w.
    - Hence the graph G is a connected acyclic graph (i.e., tree) with n vertices and n edges. Contradiction to Theorem 2.

Trees



- **Basics**
  - **Theorem 3**:

    An acyclic graph G with n vertices and $n-1$ edges is connected

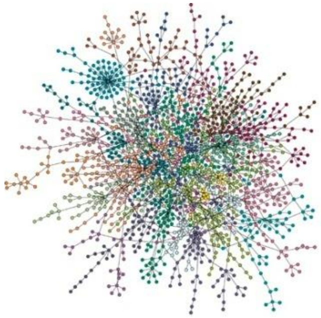
  - **Theorem 4**:

    Every connected graph G with n vertices and $n-1$ edges is tree

  - **Theorem 5**:

    A graph is a tree if it is minimally connected

Trees

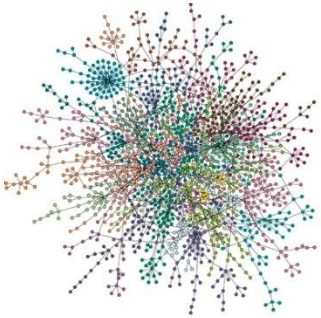


- **Basics**
  - <u>Tree Requirements</u>:

    A connected graph G with n vertices is tree:

    - If it does not contain cycles
    - If there exist a unique path between any arbitrary pair of vertices
    - If every edge is a bridge
    - If it is composed by $n - 1$ edges
    - If it contains at least two vertices of degree 1 ($n \geq 2$)
    - If it is produced only on cycle by adding an edge.

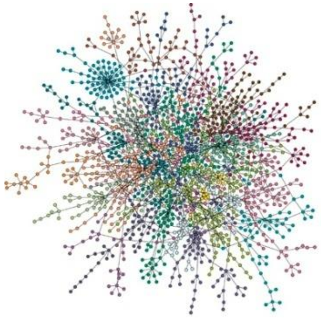
Trees



- **Basics**
  - **Corollary 3**:

    A forest (of trees) with n vertices and k components (trees) has **???** edges

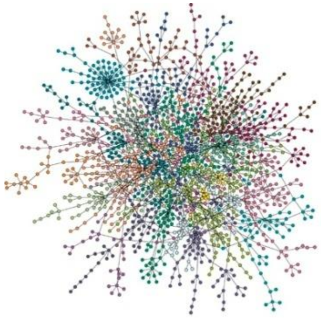
Trees



- **Basics**
  - **Corollary 3**:

    A forest (of trees) with n vertices and k components (trees) has $n - k$ edges

Trees



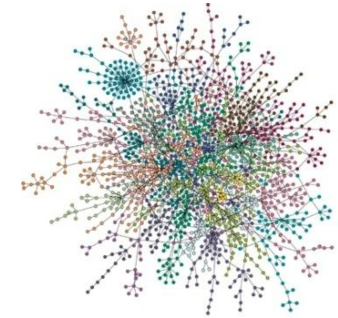
Basics

- **<u>Corollary 3</u>**:
  A forest (of trees) with n vertices and k components (trees) has $n - k$ edges

- **<u>Theorem 6 (Jordan, 1869)</u>**:
  A tree has center consisted by one or two vertices.

- **<u>Corollary 4</u>**:
  If the center of a tree is consisted by two vertices, then they are adjacent and are called <u>bicenters</u>.

TREES



Basics

- **Corollary 3**:

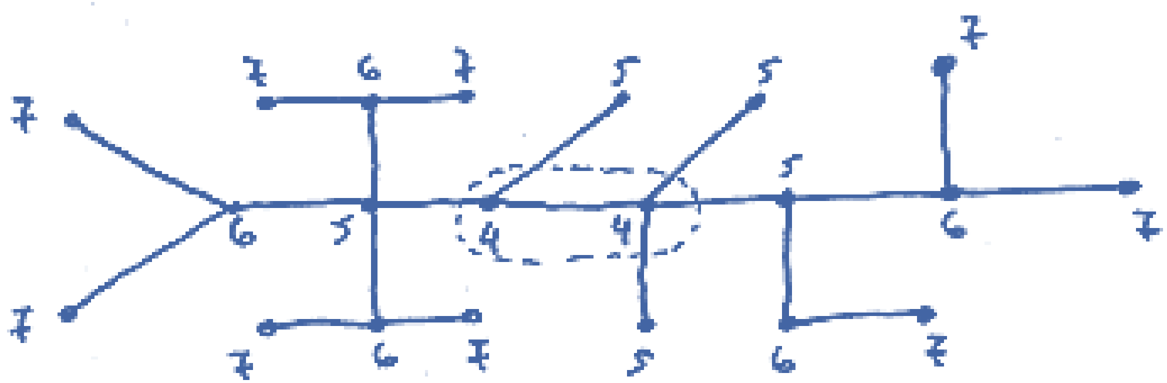
  A forest (of trees) with n vertices and k components (trees) has $n - k$ edges

- **Theorem 6 (Jordan, 1869)**:

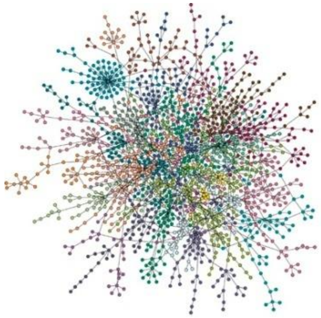
  A tree has center consisted by one or two vertices.

- **Corollary 4**:

  If the center of a tree is consisted by two vertices, then they are adjacent and are called bicenters.



Trees



Basics

- **Corollary 3**:

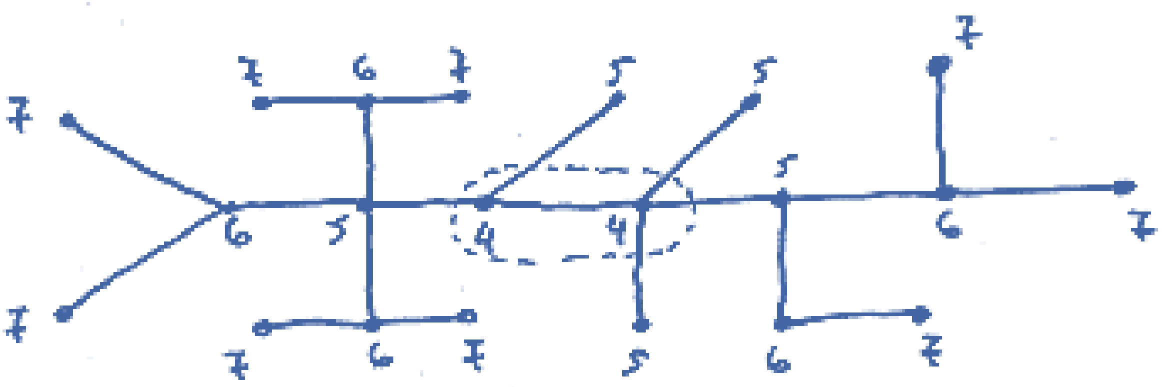
  A forest (of trees) with n vertices and k components (trees) has $n - k$ edges

- **Theorem 6 (Jordan, 1869)**:

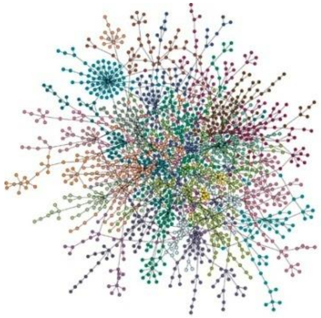
  A tree has center consisted by one or two vertices.

- **Corollary 4**:

  If the center of a tree is consisted by two vertices, then they are adjacent and are called bicenters.



TREES

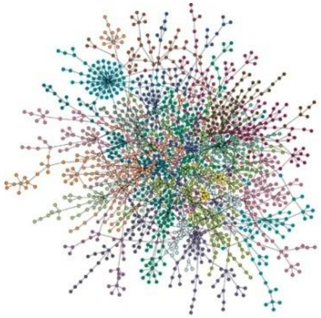


Listing of Trees

- **Theorem 8**:

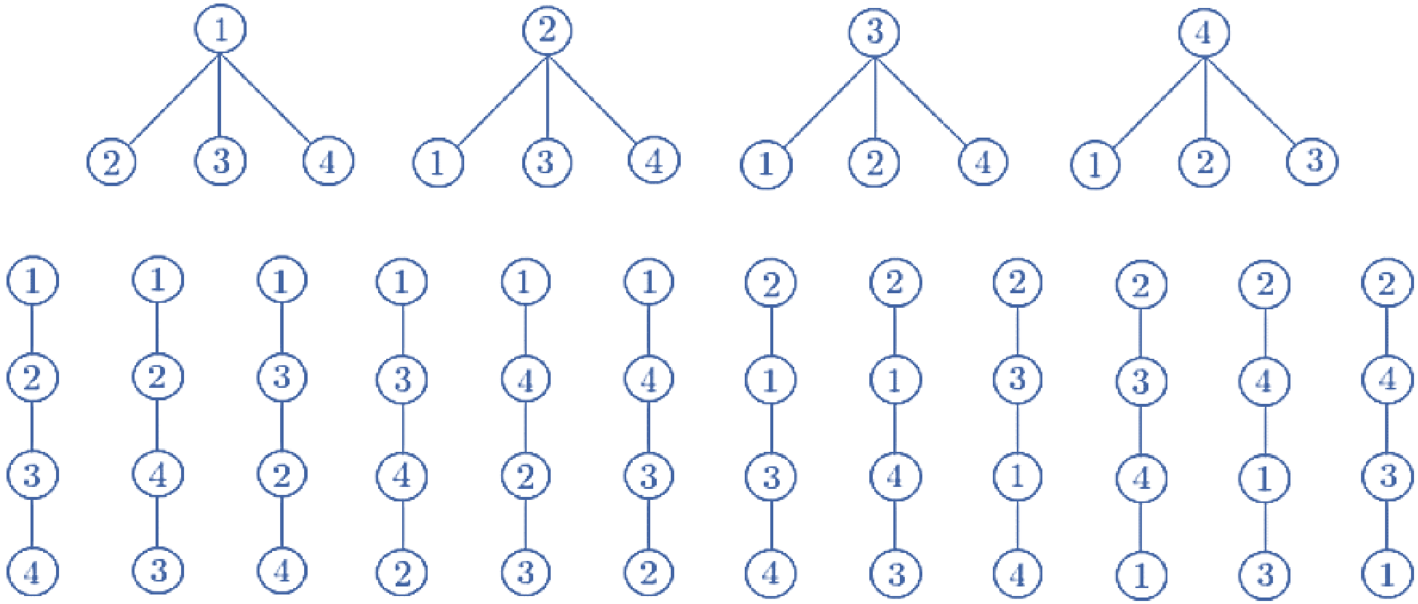
  Suppose the sum of the positive integers $d_1, d_2, \dots, d_n$ (where $n \geq 2$) is $2n - 2$. The number of trees with n nodes, where node degrees are $d_1, d_2, \dots, d_n$, equals: $\frac{(n-2)!}{(d_1-1)!(d_2-1)!\dots(d_n-1)!}$

TREES

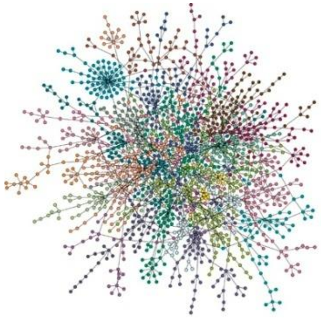


- **Listing of Trees**
  - **Theorem 8**:

    The number of discrete trees with labeled vertices, of order n, are n^{n-2}



Trees



Listing of Trees

- <u>Theorem 8</u>:

  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

Arthur Cayley
1857

$C_k\ H_{2k+2}$, number of isomers

Carbon : 4 bonds

Hydrogen: 1 bond

$$n = k + 2k + 2 = 3k + 2$$

$$m = \frac{\Sigma d\,(v)}{2} = \frac{4k + 2k + 2}{2} = 3k + 1$$

Trees

- **Listing of Trees**
  - **Theorem 8**:

    The number of discrete trees with labeled vertices, of order n, are n^{n-2}

Arthur Cayley
1857

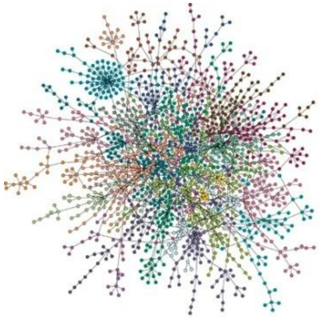
$C_k\, H_{2k+2}$, number of isomers

Carbon : 4 bonds

Hydrogen: 1 bond

$$n = k + 2k + 2 = 3k + 2$$

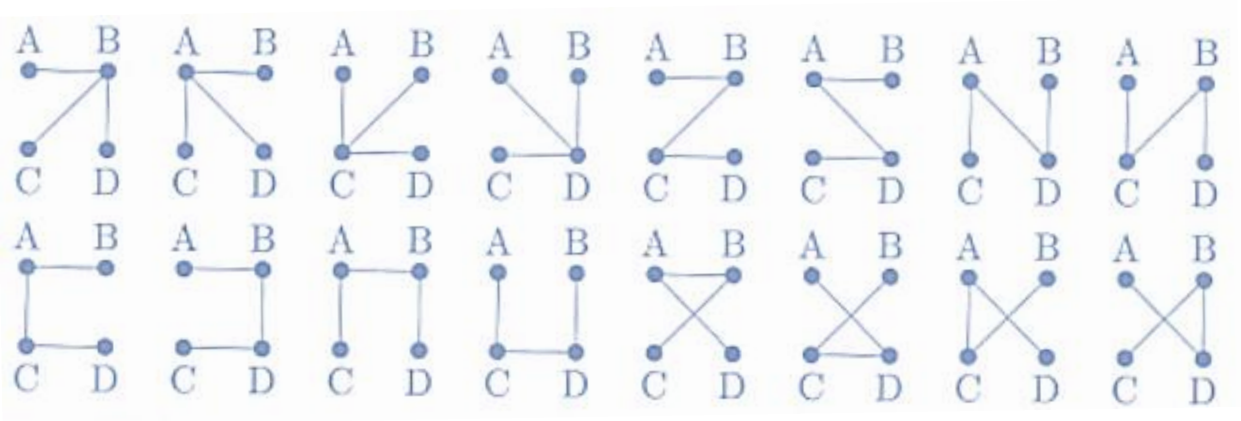
$$m = \frac{\Sigma d\,(v)}{2} = \frac{4k + 2k + 2}{2} = 3k + 1$$



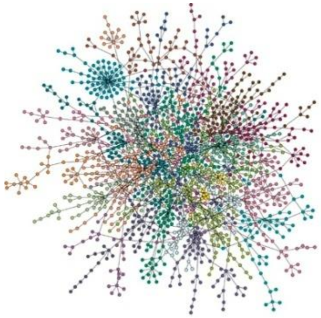
TREES

- **Listing of Trees**
  - **Theorem 8**:

    The number of discrete trees with labeled vertices, of order n, are n^{n-2}



TREES

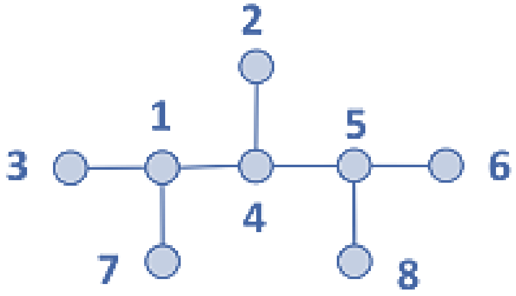


Listing of Trees

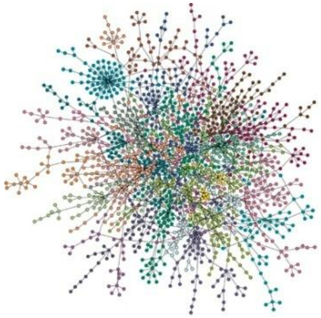
- **Theorem 8**:

  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

  1. Assign labels to the vertices of the tree as $1, 2, \dots, n$.
  2. Find the vertex with degree 1 with the smallest inscription, let a_1 and delete it.
  3. Let b_1 its neighboring vertex
  4. Repeat the procedure in the remaining subgraph.
  5. After $n-2$ deletions, the resulting tree has one edge, having created S: $(b_1, b_2, \dots, b_{n-2})$.



TREES

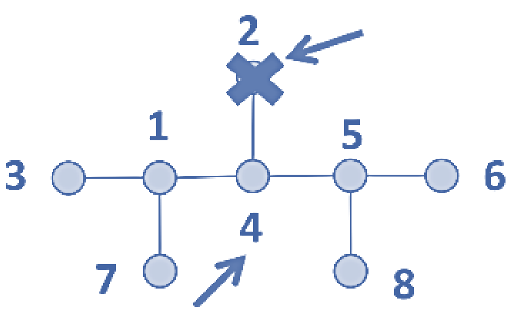


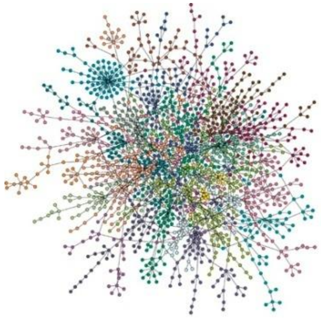
Listing of Trees

- **Theorem 8**:

  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

  1. Assign labels to the vertices of the tree as $1, 2, \dots, n$.
  2. Find the vertex with degree 1 with the smallest inscription, let a_1 and delete it.
  3. Let b_1 its neighboring vertex
  4. Repeat the procedure in the remaining subgraph.
  5. After $n-2$ deletions, the resulting tree has one edge, having created $S\colon (b_1, b_2, \dots, b_{n-2})$.



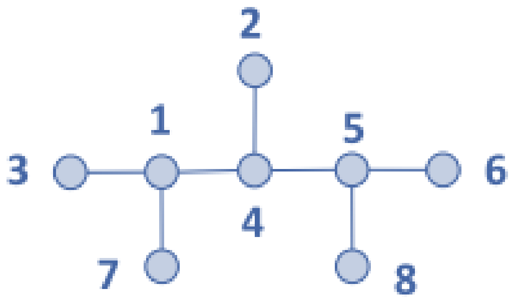


Listing of Trees

- **Theorem 8**:

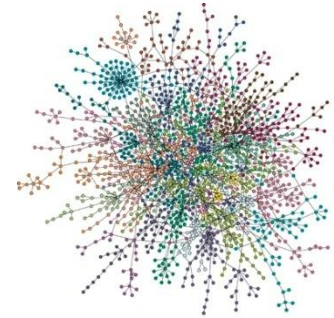
  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

  1. Assign labels to the vertices of the tree as $1, 2, \dots, n$.
  2. Find the vertex with degree 1 with the smallest inscription, let a_1 and delete it.
  3. Let b_1 its neighboring vertex
  4. Repeat the procedure in the remaining subgraph.
  5. After $n-2$ deletions, the resulting tree has one edge, having created S: $(b_1, b_2, \dots, b_{n-2})$.



$\Rightarrow$ $S = ()$

Trees

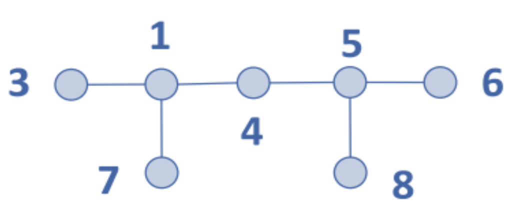


Listing of Trees

- **Theorem 8**:

  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

  1. Assign labels to the vertices of the tree as $1, 2, \dots, n$.
  2. Find the vertex with degree 1 with the smallest inscription, let a_1 and delete it.
  3. Let b_1 its neighboring vertex
  4. Repeat the procedure in the remaining subgraph.
  5. After $n-2$ deletions, the resulting tree has one edge, having created S: $(b_1, b_2, \dots, b_{n-2})$.



$\Rightarrow$ S = (4)

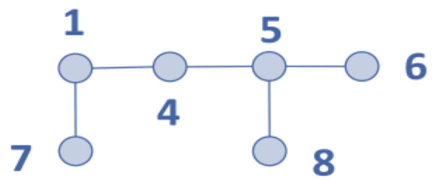
TREES

Listing of Trees

- **Theorem 8**:

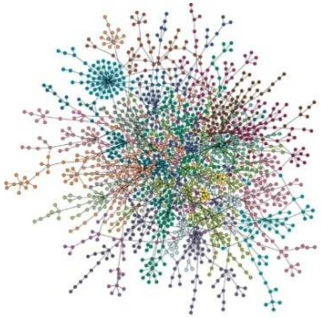
  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

1. Assign labels to the vertices of the tree as $1, 2, \dots, n$.
2. Find the vertex with degree 1 with the smallest inscription, let a_1 and delete it.
3. Let b_1 its neighboring vertex
4. Repeat the procedure in the remaining subgraph.
5. After $n - 2$ deletions, the resulting tree has one edge, having created $S\colon (b_1, b_2, \dots, b_{n-2})$.



$\Rightarrow$ $S = (4, 1)$

TREES

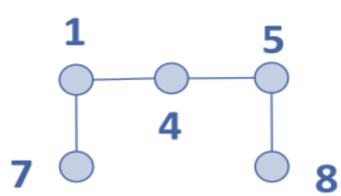


Listing of Trees

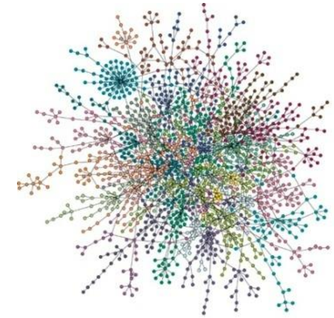
- <u>**Theorem 8**</u>:

  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

  1. Assign labels to the vertices of the tree as $1, 2, \dots, n$.
  2. Find the vertex with degree 1 with the smallest inscription, let a_1 and delete it.
  3. Let b_1 its neighboring vertex
  4. Repeat the procedure in the remaining subgraph.
  5. After $n - 2$ deletions, the resulting tree has one edge, having created S: $(b_1, b_2, \dots, b_{n-2})$.



$\Rightarrow$ **S = (4, 1, 5)**



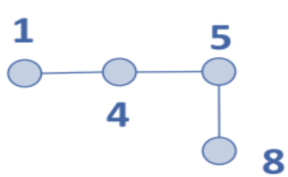
Trees

Listing of Trees

- **<u>Theorem 8</u>**:

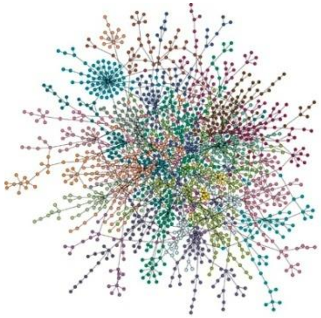
  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

  1. Assign labels to the vertices of the tree as $1, 2, \dots, n$.
  2. Find the vertex with degree 1 with the smallest inscription, let a_1 and delete it.
  3. Let b_1 its neighboring vertex
  4. Repeat the procedure in the remaining subgraph.
  5. After $n-2$ deletions, the resulting tree has one edge, having created $S\!:\ (b_1, b_2, \dots, b_{n-2})$.



$\Rightarrow$ **S = (4, 1, 5, 1)**

Trees

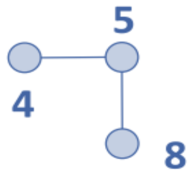


Listing of Trees

- **Theorem 8**:

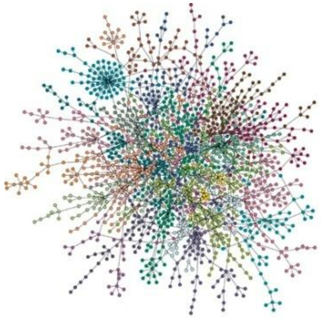
  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

  1. Assign labels to the vertices of the tree as $1, 2, \dots, n$.
  2. Find the vertex with degree 1 with the smallest inscription, let a_1 and delete it.
  3. Let b_1 its neighboring vertex
  4. Repeat the procedure in the remaining subgraph.
  5. After $n-2$ deletions, the resulting tree has one edge, having created $S\colon\ (b_1, b_2, \dots, b_{n-2})$.



$\Rightarrow$ **S = (4, 1, 5, 1, 4)**

Trees



Listing of Trees

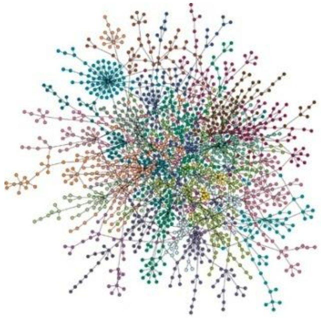
- **Theorem 8**:

  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

  1. Assign labels to the vertices of the tree as $1, 2, \dots, n$.
  2. Find the vertex with degree 1 with the smallest inscription, let a_1 and delete it.
  3. Let b_1 its neighboring vertex
  4. Repeat the procedure in the remaining subgraph.
  5. After $n-2$ deletions, the resulting tree has one edge, having created S: $(b_1, b_2, \dots, b_{n-2})$.

$\Rightarrow$ **S = (4, 1, 5, 1, 4, 5)**

TREES



Listing of Trees

- **Theorem 8**:

  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

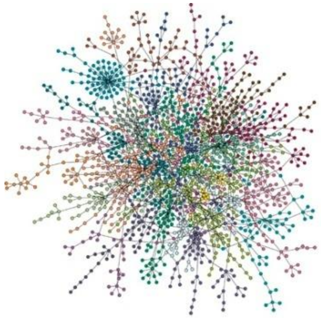
- **Prufer Encoding**

  Input: A tree T with numerical labeling on its vertices a_i $(1 \leq i \leq n)$

  Output: A Prufer sequence of length $n-2$

  1) For $i = 1:1:n-2$
  2) Let v the vertex with the minimum label
  3) Let b_i the label of the only neighbor of vertex v
  4) $T \leftarrow T - v$
  5) Return $S(b_1, b_2, \dots\ b_{n-2})$

TREES

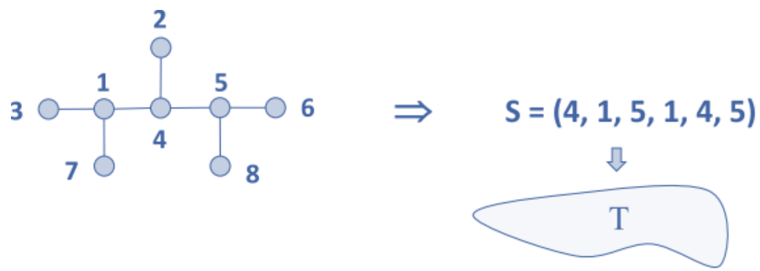


Listing of Trees

- **Theorem 8**:
  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

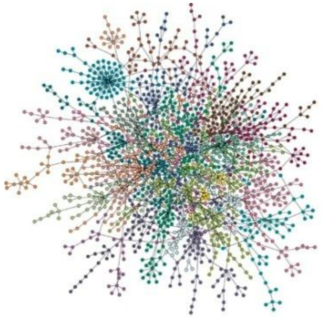
- We can build a tree T in a unique way from $S = (b_1, b_2, \dots, b_{n-2})$ containing non-pending vertices (vertices of degree 1).

- Each element of $S = (b_1, b_2, \dots, b_{n-2})$ may take values $1 \leq \ b_i \leq n$ (where $1 \ \leq \ i \ \leq \ n - 2) \ \Rightarrow n^{n-2}$.



- Find the less number in range $[1, n]$ that does not appear in S (Prufer Decoding).

Trees



Listing of Trees

- **Theorem 8**:
  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

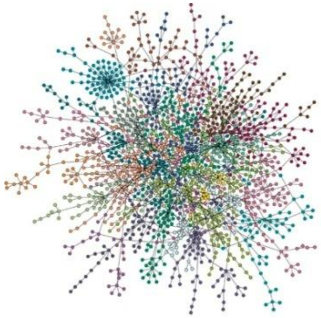
- We can build a tree T in a unique way from $S = (b_1, b_2, \dots, b_{n-2})$ containing non-pending vertices (vertices of degree 1).

1. Let list $L = (1, 2, \dots, n)$.
2. Select from L the minimum label, let l_1, which does not belong to S.
3. The edge (l_1, s_1) belongs to T.
4. Delete l_1 from L and s_1 of S.
5. Repeat with the new sequences L and S.

L = (1, 2, 3, 4, 5, 6, 7, 8)

S = (4, 1, 5, 1, 4, 5)

TREES



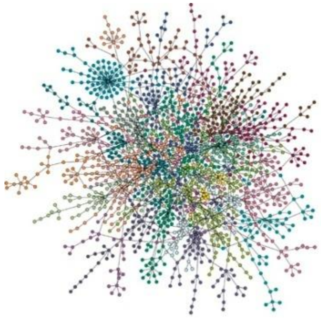
Listing of Trees

- **Theorem 8**:
  The number of discrete trees with labeled vertices, of order n, are n^{n-2}
- We can build a tree T in a unique way from $S = (b_1, b_2, \dots, b_{n-2})$ containing non-pending vertices (vertices of degree 1).

1. Let list $L = (1, 2, \dots, n)$.
2. Select from L the minimum label, let l_1, which does not belong to S.
3. The edge (l_1, s_1) belongs to T.
4. Delete l_1 from L and s_1 of S.
5. Repeat with the new sequences L and S.

L = (1, 2, 3, 4, 5, 6, 7, 8)

S = (4, 1, 5, 1, 4, 5)



Listing of Trees

- **Theorem 8**:
  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

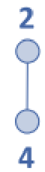
- We can build a tree T in a unique way from $S = (b_1, b_2, \ldots, b_{n-2})$ containing non-pending vertices (vertices of degree 1).

1. Let list $L = (1, 2, \ldots, n)$.
2. Select from L the minimum label, let l_1, which does not belong to S.
3. The edge (l_1, s_1) belongs to T.
4. Delete l_1 from L and s_1 of S.
5. Repeat with the new sequences L and S.

L = (1, 2, 3, 4, 5, 6, 7, 8)

S = (4, 1, 5, 1, 4, 5)

(2,4) -



Trees

Listing of Trees

- **Theorem 8**:

  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

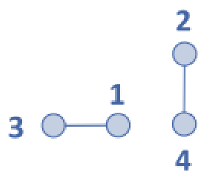
- We can build a tree T in a unique way from $S = (b_1, b_2, \ldots, b_{n-2})$ containing non-pending vertices (vertices of degree 1).

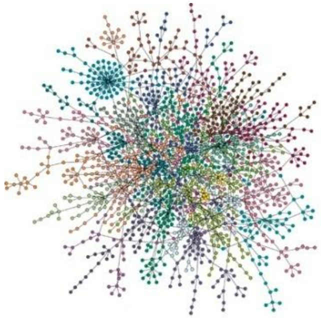
1. Let list $L = (1, 2, \ldots, n)$.
2. Select from L the minimum label, let l_1, which does not belong to S.
3. The edge (l_1, s_1) belongs to T.
4. Delete l_1 from L and s_1 of S.
5. Repeat with the new sequences L and S.

L = (1, x, 3, 4, 5, 6, 7, 8)

S = (x, 1, 5, 1, 4, 5)

(2,4) - (3,1)





Listing of Trees

- **Theorem 8**:
  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

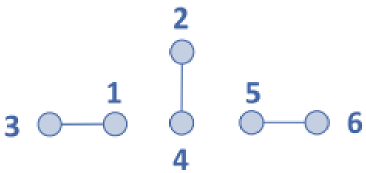
- We can build a tree T in a unique way from $S = (b_1, b_2, \ldots, b_{n-2})$ containing non-pending vertices (vertices of degree 1).

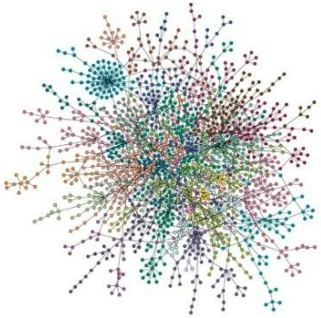
1. Let list $L = (1, 2, \ldots, n)$.
2. Select from L the minimum label, let l_1, which does not belong to S.
3. The edge (l_1, s_1) belongs to T.
4. Delete l_1 from L and s_1 of S.
5. Repeat with the new sequences L and S.

L = (1, x, x, 4, 5, 6, 7, 8)

S = (x, x, 5, 1, 4, 5)

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





Trees

Listing of Trees

- **Theorem 8**:
  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

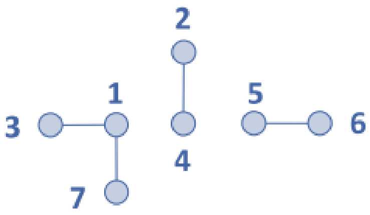
- We can build a tree T in a unique way from $S = (b_1, b_2, \dots, b_{n-2})$ containing non-pending vertices (vertices of degree 1).

1. Let list $L = (1, 2, \dots, n)$.
2. Select from L the minimum label, let l_1, which does not belong to S.
3. The edge (l_1, s_1) belongs to T.
4. Delete l_1 from L and s_1 of S.
5. Repeat with the new sequences L and S.

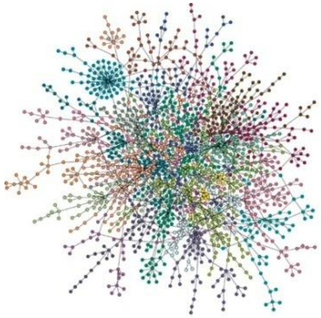
L = (1, x, x, 4, 5, x, 7, 8)

S = (x, x, x, 1, 4, 5)

(2,4) - (3,1) - (6,5) - (7,1)



Trees



Listing of Trees

- **Theorem 8**:

  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

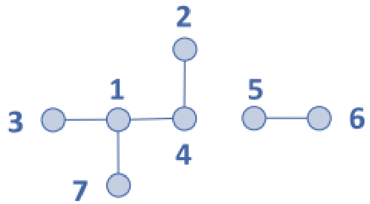
- We can build a tree T in a unique way from $S = (b_1, b_2, \dots, b_{n-2})$ containing non-pending vertices (vertices of degree 1).

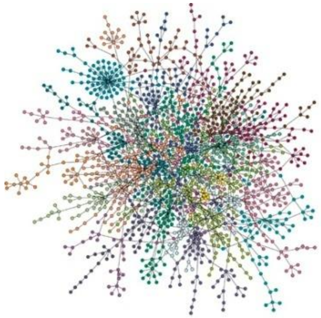
1. Let list $L = (1, 2, \dots, n)$.
2. Select from L the minimum label, let l_1, which does not belong to S.
3. The edge (l_1, s_1) belongs to T.
4. Delete l_1 from L and s_1 of S.
5. Repeat with the new sequences L and S.

L = (1, x, x, 4, 5, x, x, 8)

S = (x, x, x, x, 4, 5)

(2,4) - (3,1) - (6,5) - (7,1) - (1,4)





Listing of Trees

- **Theorem 8**:
  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

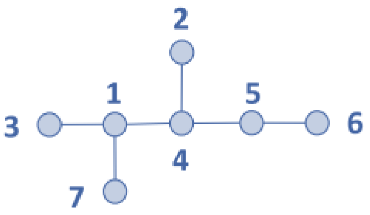
- We can build a tree T in a unique way from $S = (b_1, b_2, \ldots, b_{n-2})$ containing non-pending vertices (vertices of degree 1).

  1. Let list $L = (1, 2, \ldots, n)$.
  2. Select from L the minimum label, let l_1, which does not belong to S.
  3. The edge (l_1, s_1) belongs to T.
  4. Delete l_1 from L and s_1 of S.
  5. Repeat with the new sequences L and S.

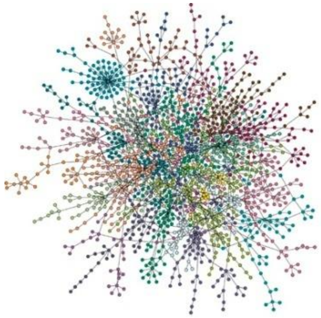
L = (x, x, x, 4, 5, x, x, 8)

S = (x, x, x, x, x, 5)

(2,4) - (3,1) - (6,5) - (7,1) - (1,4) - (4,5)



Trees



Listing of Trees

- **Theorem 8**:
  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

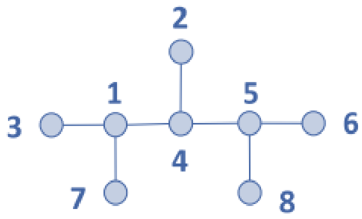
- We can build a tree T in a unique way from $S = (b_1, b_2, \dots, b_{n-2})$ containing non-pending vertices (vertices of degree 1).

1. Let list $L = (1, 2, \dots, n)$.
2. Select from L the minimum label, let l_1, which does not belong to S.
3. The edge (l_1, s_1) belongs to T.
4. Delete l_1 from L and s_1 of S.
5. Repeat with the new sequences L and S.

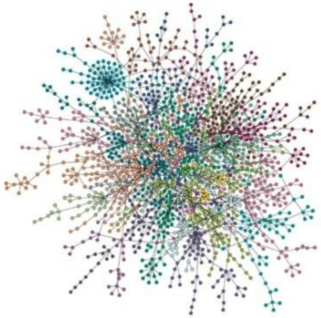
L = (x, x, x, x, 5, x, x, 8)

S = (x, x, x, x, x, x)

(2,4) - (3,1) - (6,5) - (7,1) - (1,4) - (4,5) - (5,8)



TREES



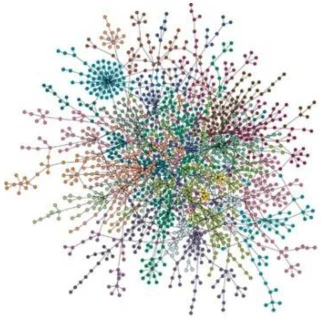
Listing of Trees

- **Theorem 8**:
  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

- We can build a tree T in a unique way from $S = (b_1, b_2, \dots, b_{n-2})$ containing non-pending vertices (vertices of degree 1).

- **Prufer Decoding**
  Input: A Prufer sequence of length $n - 2$
  Output: A tree T with numerical labeling on its vertices a_i $(1 \leq i \leq n)$
  1) Initialize empty list P (Prufer sequence)
  2) Initialize list $L = 1,2, \dots, n$
  3) $F \leftarrow$ forest from n independent vertices enumerated from 1 to n
  4) For $i = 1{:}\,1{:}\,n - 2$
     a. Let k the minimum number in $L : \notin \ P$
     b. Let j the first number in P
     c. Connect by an edge the vertices k and j
     d. Delete k from L
     e. Delete the first appearance of j from P
  5) Connect by an edge the vertices with the remaining numbers of P
  6) Return the forest (tree)F

Trees



Listing of Trees

- **<u>Theorem 8</u>**:

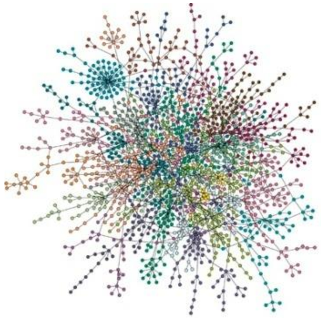
  The number of discrete trees with labeled vertices, of order n, are n^{n-2}

- **Corollary**:

  The number of discrete rooted trees with labeled vertices, of order n, are n^{n-1}

  - In a rooted tree with labeled vertices, a vertex is defined to be the root.
  - A non-rooted tree is also called free.
  - For each one of the n^{n-2} discrete trees with labeled vertices there are produced n different rooted trees, as any vertex can be defined as root.

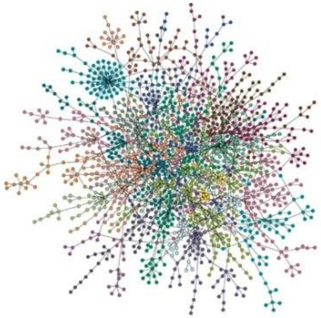
TREES



Spanning Trees

- From each connected graph $G(V, E)$ can be produced $2^{|E|}$ different subgraphs. Many of these graphs are trees.
- **Spanning Tree** of a connected graph G is called the tree T that is subgraph of G and it holds that $V(T) = V(G)$.
- A spanning Tree is also called Skeleton, or Scaffolding, or Maximal Tree of G.
- A non-connected graph G of k components has a **Spanning Forest** of k Spanning Trees.
- The edges of a Spanning Tree are called **Branches**.
- The vertices of G that are not Branches of its Spanning Tree are called **Chords**.

Trees

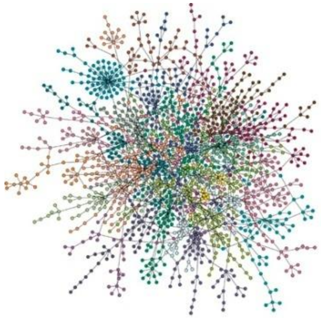


- **Spanning Trees**
  - **<u>Theorem 9</u>**:

    Every connected graph has at least one Spanning Tree.

    - If the connected graph is a Tree then the Spanning Tree is the graph.
    - Otherwise, a Spanning Tree can be constructed by deleting sequentially edges that belong to cycles until only bridges are left.
    - The number of the deleted edges equals the number of chords of the graph.

Trees



Spanning Trees

- **Theorem 9**:

  Every connected graph has at least one Spanning Tree.

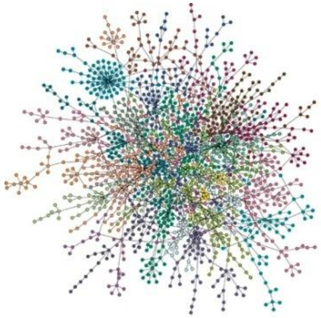
  - If the connected graph is a Tree then the Spanning Tree is the graph.
  - Otherwise, a Spanning Tree can be constructed by deleting sequentially edges that belong to cycles until only bridges are left.
  - The number of the deleted edges equals the number of chords of the graph.

- **Corollary**:

  Every connected graph G of n vertices and m edges can be considered as the union of a Spanning Tree T of $n-1$ branches and a subgraph $\check{T}$ of $m-n+1$ chords.

  - Subgraph $\check{T}$ is the complement of Tree T by the graph G, and it is called **Chord Set** or **CoTree**. Hence, it holds that $G = T \cup \check{T}$.

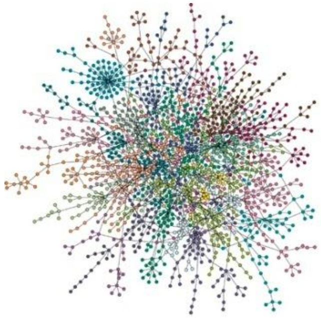
Trees



- **Spanning Trees**
  - **Theorem 10**:

    If in a graph G it holds that $d(G) \geq k$, and the graph T is a tree of $k+1$ vertices , then T is a Spanning Tree of G

Trees

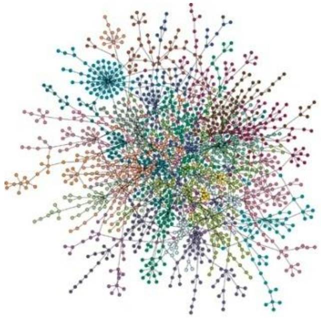


- **Spanning Trees**
  - **Theorem 10**:

    If in a graph G it holds that $d(G) \geq k$, and the graph T is a tree of $k+1$ vertices , then T is a Spanning Tree of G

    Inductively on k …

TREES



Spanning Trees

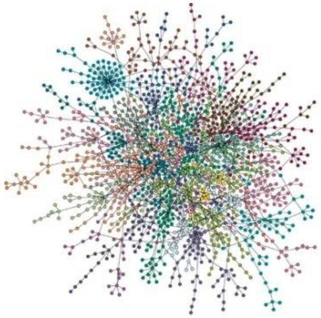
- **Theorem 10**:

  If in a graph G it holds that $d(G) \geq k$, and the graph T is a tree of $k+1$ vertices , then T is a Spanning Tree of G

  Inductively on k …

  - If $k = 0$, then $T = K_1$ is a subgraph of each graph.

TREES



Spanning Trees

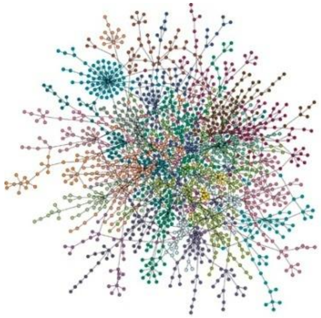
- **Theorem 10**:

  If in a graph G it holds that $d(G) \geq k$, and the graph T is a tree of $k+1$ vertices , then T is a Spanning Tree of G

  Inductively on k …

  - If $k = 0$, then $T = K_1$ is a subgraph of each graph.
  - If $k = 1$, then $T = K_2$ is a subgraph of each non-empty graph

TREES



Spanning Trees

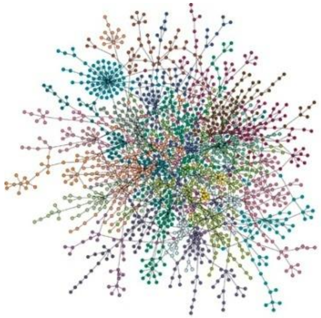
- **Theorem 10**:

  If in a graph G it holds that $d(G) \geq k$, and the graph T is a tree of $k+1$ vertices , then T is a Spanning Tree of G

  Inductively on k …

  - If $k = 0$, then $T = K_1$ is a subgraph of each graph.
  - If $k = 1$, then $T = K_2$ is a subgraph of each non-empty graph
  - Let that the Theorem holds for every tree T_1 of k vertices and every graph of vertices of minimum degree $k-1$. We will prove the truth for every tree T of k_1 vertices and every graph G with vertices of minimum degree k.

TREES



Spanning Trees

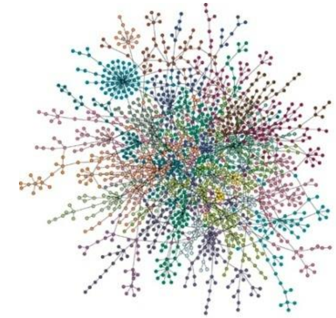
- **Theorem 10**:

  If in a graph G it holds that $d(G) \geq k$, and the graph T is a tree of $k+1$ vertices , then T is a Spanning Tree of G

  Inductively on k …

  - If $k = 0$, then $T = K_1$ is a subgraph of each graph.
  - If $k = 1$, then $T = K_2$ is a subgraph of each non-empty graph
  - Let that the Theorem holds for every tree T_1 of k vertices and every graph of vertices of minimum degree $k - 1$. We will prove the truth for every tree T of k_1 vertices and every graph G with vertices of minimum degree k.
    - Let v a vertex of degree 1, of tree T.
    - Let that vertex v is adjacent to vertex $w \in T$

Trees



Spanning Trees

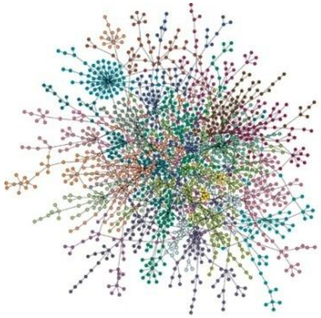
- **Theorem 10**:

  If in a graph G it holds that $d(G) \geq k$, and the graph T is a tree of $k + 1$ vertices , then T is a Spanning Tree of G

  Inductively on k …

  - If $k = 0$, then $T = K_1$ is a subgraph of each graph.
  - If $k = 1$, then $T = K_2$ is a subgraph of each non-empty graph
  - Let that the Theorem holds for every tree T_1 of k vertices and every graph of vertices of minimum degree $k - 1$. We will prove the truth for every tree T of k_1 vertices and every graph G with vertices of minimum degree k.
    - Since $T - v$ is a tree of k vertices, and the vertices of graph $G - v$ have minimum degree greater or equal to $k - 1$, from the assumption of the induction it results that $T - v \subseteq G - v \subseteq G$.

TREES



Spanning Trees

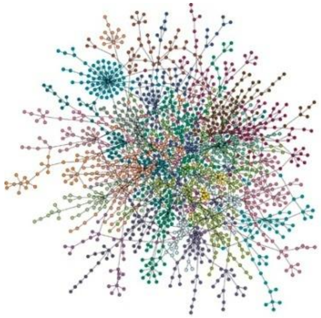
- **Theorem 10**:

  If in a graph G it holds that $d(G) \geq k$, and the graph T is a tree of $k+1$ vertices , then T is a Spanning Tree of G

  Inductively on k …

  - If $k = 0$, then $T = K_1$ is a subgraph of each graph.
  - If $k = 1$, then $T = K_2$ is a subgraph of each non-empty graph
  - Let that the Theorem holds for every tree T_1 of k vertices and every graph of vertices of minimum degree $k-1$. We will prove the truth for every tree T of k_1 vertices and every graph G with vertices of minimum degree k.
    - Since the degree of vertex w in graph G is at least k and the tree $T - v$ has $k-1$ vertices, it results that vertex w has an adjacent vertex in graph $G \notin V(T)$.
    - It implies that the tree T is subgraph of G.

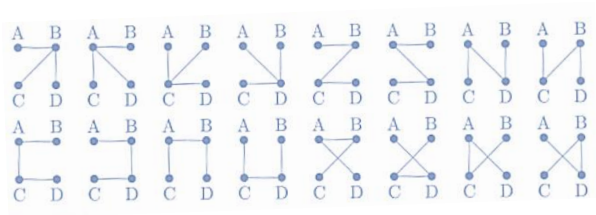
Trees



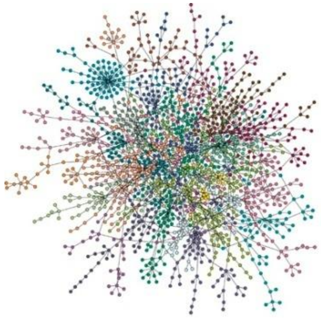
- **Spanning Trees**
  - **<u>Theorem 11</u>**:

    The number of discrete Spanning Trees of a complete graph K_n is n^{n-2}

    - For every tree with labeled vertices, of order n, there exists a unique Spanning Tree of graph K_n. Inversely, from each Spanning Tree of graph K_n results a unique tree with labeled vertices that is of order n.



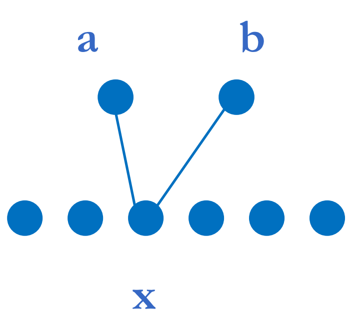
TREES



- **Spanning Trees**
  - **Theorem 12**:

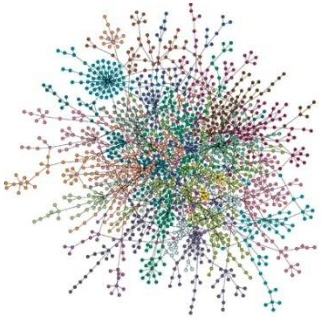
    The number of discrete Spanning Trees of a complete bipartite graph $K_{n,m}$ is $m^{n-1}n^{m-1}$.

  - $K_{2,n} \rightarrow n\, 2^{n-1}$



- The vertices a, b are connected with vertex x by n different ways.
- The rest $n-1$ vertices can be connected either with vertex a or with vertex b
- … $n\, 2^{n-1}$ discrete Spanning Trees

Trees

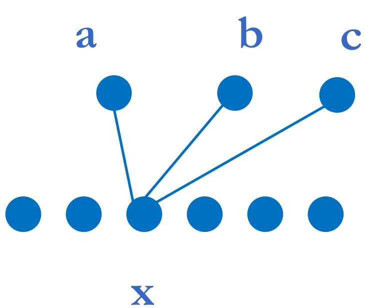


Spanning Trees

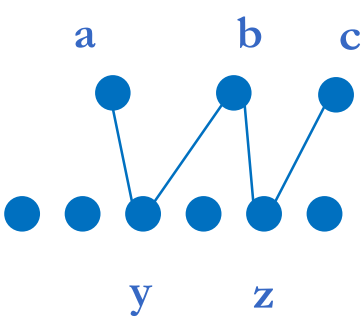
- <u>**Theorem 12**</u>:

  The number of discrete Spanning Trees of a complete bipartite graph $K_{n,m}$ is $m^{n-1}n^{m-1}$.

- $K_{3,n} \rightarrow n^2\, 3^{n-1}$



<u>Paths of length 2</u>

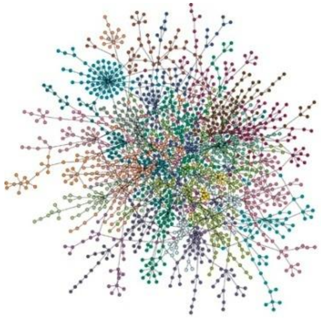


<u>Paths of length 4</u>

  - <u>There exists 6 cases with a,b,c and y,z.</u>
  - The vertices y and z can be selected by $\binom{n}{2} = \frac{n(n-1)}{2}$ ways.
  - (a) For paths of length 2: $n\, 3^{n-1}$
  - (b) For paths of length 4: $6\, \frac{n(n-1)}{2}\, 3^{n-2}$

From $(a) + (b) \rightarrow\ n(n-1)3^{n-1}$

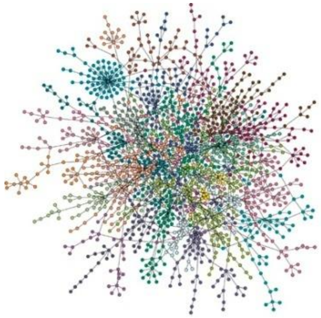
Trees



- **Spanning Trees**
  - **<u>Theorem 12</u>**:

    The number of discrete Spanning Trees of a complete bipartite graph $K_{n,m}$ is ???.

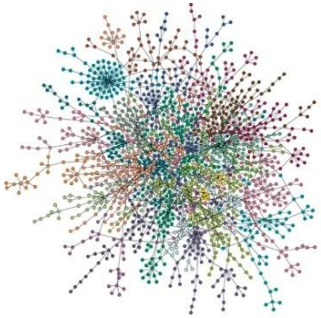
TREES



- **Spanning Trees**
  - **Theorem 12**:

    The number of discrete Spanning Trees of a complete bipartite graph $K_{n,m}$ is $m^{n-1}n^{m-1}$.

Trees



Spanning Trees

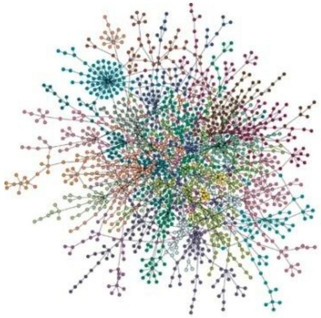
- **Theorem 12**:

  The number of discrete Spanning Trees of a complete bipartite graph $K_{n,m}$ is $m^{n-1}n^{m-1}$.

- **Theorem 13**:

  The number of discrete Spanning Trees of a wheel graph W_n, is $\left(\frac{3+\sqrt{5}}{2}\right)^{n-1} + \left(\frac{3-\sqrt{5}}{2}\right)^{n-1} - 2$

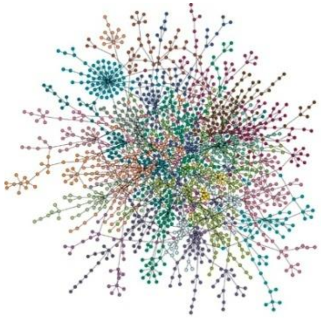
Trees



- **Spanning Trees**
  - **Theorem 14 (Matrix-tree theorem) - Kirchoff**:
    - $A \rightarrow$ Adjacency Matrix
    - $C \rightarrow$ Degree Matrix
    - $C - A \rightarrow$ Admittance Matrix
    - $B_{ij} = (C - A)_{ij} \rightarrow$ Minor Matrix
    - $(-1)^{i+j}\,|B_{ij}| \rightarrow$ CoFactor

    The number of genetic trees equals the CoFactor

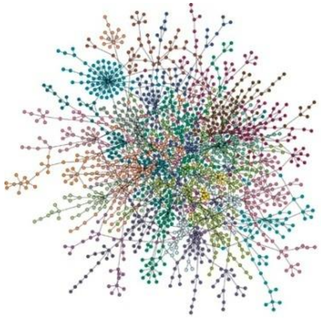
Trees



- **Spanning Trees**
  - **Theorem 14 (Matrix-tree theorem) - Kirchoff**:
    - $A \rightarrow$ Adjacency Matrix
    - $C \rightarrow$ Degree Matrix
      - $C\ (i,j) = 0\ for\ i \neq j\ and\ C(i,i)\ =\ d(v_i)\ for\ 1 \leq i \leq n.$
    - $C - A \rightarrow$ Admittance Matrix
    - $B_{ij} = (C - A)_{ij} \rightarrow$ Minor Matrix
    - $(-1)^{i+j}\ |B_{ij}| \rightarrow$ CoFactor

The number of genetic trees equals the CoFactor

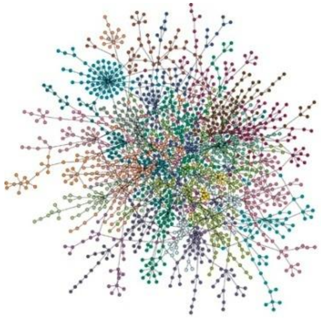
TREES



- **Spanning Trees**
  - **Theorem 14 (Matrix-tree theorem) - Kirchoff**:
    - $A \rightarrow$ Adjacency Matrix
    - $C \rightarrow$ Degree Matrix
    - $C - A \rightarrow$ Admittance Matrix
    - $B_{ij} = (C - A)_{ij} \rightarrow$ Minor Matrix
      - If from a two-dimensional table B with $n \times n$ elements delete the $i - th$ line and the $j - th$ column, then a matrix B_{ij} is called Minor Matrix in position i, j.
    - $(-1)^{i+j} \left|B_{ij}\right| \rightarrow$ CoFactor

The number of genetic trees equals the CoFactor

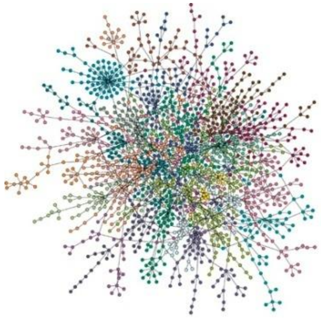
Trees



- **Spanning Trees**
  - **Theorem 14 (Matrix-tree theorem) - Kirchoff**:
    - $A \rightarrow$ Adjacency Matrix
    - $C \rightarrow$ Degree Matrix
    - $C - A \rightarrow$ Admittance Matrix
    - $B_{ij} = (C - A)_{ij} \rightarrow$ Minor Matrix
    - $(-1)^{i+j}\,|B_{ij}| \rightarrow$ CoFactor
      - Cofactor of matrix B in position i, j is called the value $(-1)^{i+j}\,|B_{ij}|$

    The number of genetic trees equals the CoFactor

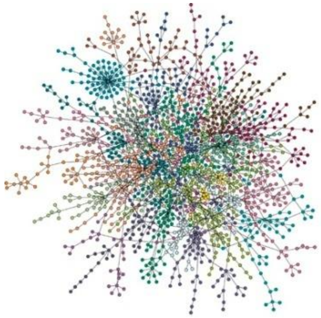
TREES



- **Spanning Trees**
  - **Theorem 14 (Matrix-tree theorem) - Kirchoff**:
    - $A \rightarrow$ Adjacency Matrix
    - $C \rightarrow$ Degree Matrix
    - $C - A \rightarrow$ Admittance Matrix
    - $B_{ij} = (C - A)_{ij} \rightarrow$ Minor Matrix
    - $(-1)^{i+j}\,|B_{ij}| \rightarrow$ CoFactor

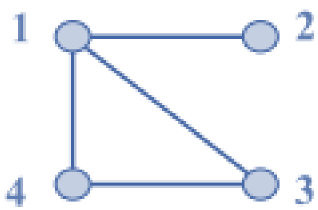
    The number of genetic trees equals the CoFactor

Trees



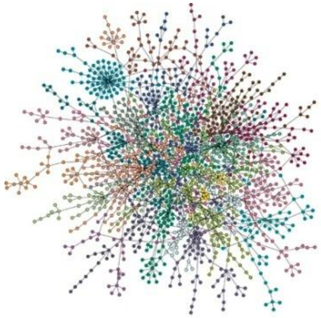
Spanning Trees

- **Theorem 14 (Matrix-tree theorem) - Kirchoff**:
  - $A \rightarrow$ Adjacency Matrix
  - $C \rightarrow$ Degree Matrix
  - $C - A \rightarrow$ Admittance Matrix
  - $B_{ij} = (C - A)_{ij} \rightarrow$ Minor Matrix
  - $(-1)^{i+j} \left| B_{ij} \right| \rightarrow$ CoFactor

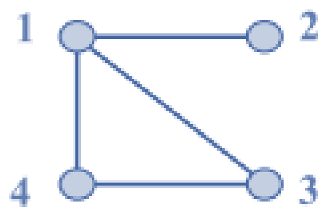


The number of genetic trees equals the CoFactor

Trees

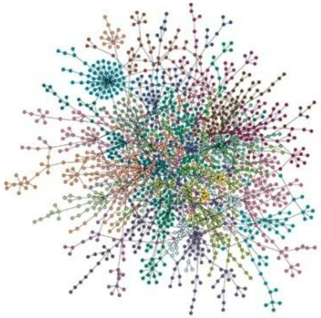


- **Spanning Trees**
  - <u>**Theorem 14 (Matrix-tree theorem) - Kirchoff**</u>:
    - $\boldsymbol{A}$ → **Adjacency Matrix**
    - C → Degree Matrix
    - $C - A$ → Admittance Matrix
    - $B_{ij} = (C - A)_{ij}$ → Minor Matrix
    - $(-1)^{i+j}\,|B_{ij}|$ → CoFactor



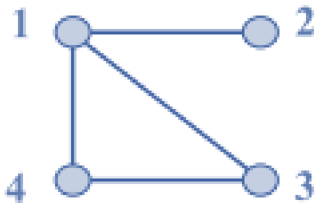
The number of genetic trees equals the CoFactor

Adjacency Matrix (A)			
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0



Spanning Trees

- **Theorem 14 (Matrix-tree theorem) - Kirchoff**:
  - $A \rightarrow$ Adjacency Matrix
  - $\boldsymbol{C} \rightarrow$ **Degree Matrix**
  - $C - A \rightarrow$ Admittance Matrix
  - $B_{ij} = (C - A)_{ij} \rightarrow$ Minor Matrix
  - $(-1)^{i+j}\,|B_{ij}| \rightarrow$ CoFactor

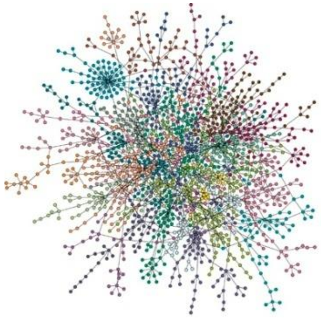


The number of genetic trees equals the CoFactor

Adjacency Matrix (A)			
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0

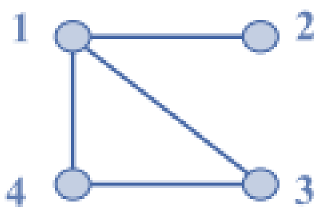
Degree Matrix (C)			
3	0	0	1
0	1	0	0
0	0	2	1
0	0	0	2

Trees



Spanning Trees

- <u>**Theorem 14 (Matrix-tree theorem) - Kirchoff**</u>:
  - $A \rightarrow$ Adjacency Matrix
  - $C \rightarrow$ Degree Matrix
  - $\boldsymbol{C - A} \rightarrow$ **Admittance Matrix**
  - $B_{ij} = (C - A)_{ij} \rightarrow$ Minor Matrix
  - $(-1)^{i+j}\,|B_{ij}| \rightarrow$ CoFactor



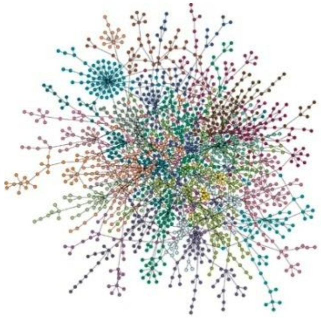
The number of genetic trees equals the CoFactor

Adjacency Matrix (A)			
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0

Degree Matrix (C)			
3	0	0	1
0	1	0	0
0	0	2	1
0	0	0	2

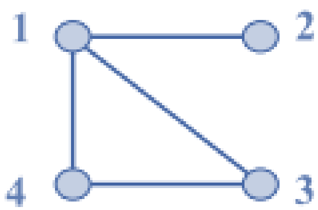
Admittance Matrix (C-A)			
3	-1	-1	-1
-1	1	0	0
-1	0	2	-1
-1	0	-1	2

TREES



Spanning Trees

- <u>**Theorem 14 (Matrix-tree theorem) - Kirchoff**</u>:
  - $A \rightarrow$ Adjacency Matrix
  - $C \rightarrow$ Degree Matrix
  - $C - A \rightarrow$ Admittance Matrix
  - $\boldsymbol{B_{ij} = (C - A)_{ij}} \rightarrow$ **Minor Matrix**
  - $(-1)^{i+j}\,|B_{ij}| \rightarrow$ CoFactor

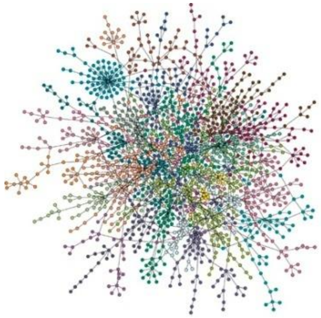


The number of genetic trees equals the CoFactor

Admittance Matrix (C-A)			
3	-1	-1	-1
-1	1	0	0
-1	0	2	-1
-1	0	-1	2

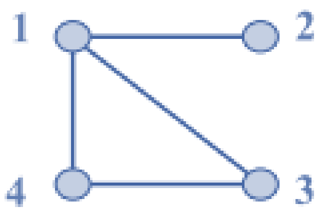
Minor Matrix (B_{11})		
1	0	0
0	2	-1
0	-1	2

Trees



Spanning Trees

- **Theorem 14 (Matrix-tree theorem) - Kirchoff**:
  - $A \rightarrow$ Adjacency Matrix
  - $C \rightarrow$ Degree Matrix
  - $C - A \rightarrow$ Admittance Matrix
  - $B_{ij} = (C - A)_{ij} \rightarrow$ Minor Matrix
  - $(-\mathbf{1})^{i+j}\,|\boldsymbol{B_{ij}}| \rightarrow$ **CoFactor**



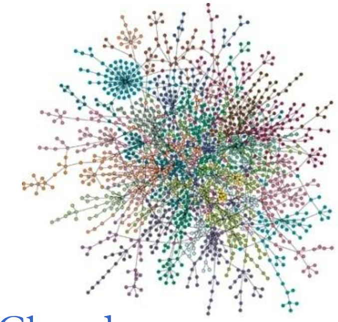
The number of genetic trees equals the CoFactor

Admittance Matrix (C-A)			
3	-1	-1	-1
-1	1	0	0
-1	0	2	-1
-1	0	-1	2

Minor Matrix (B_{11})		
1	0	0
0	2	-1
0	-1	2

$(-\mathbf{1})^2\,|\boldsymbol{B_{11}}| \rightarrow$ **CoFactor**

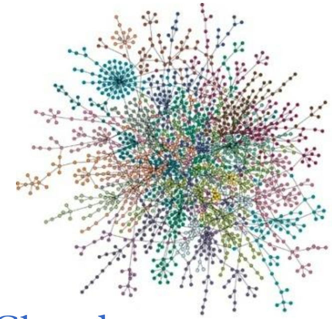
Trees



Spanning Trees

- **Fundamental Circuit**: A cycle created by a Spanning Tree and a Chord.
- Total Chords: $m - n + 1$
- $G = T \cup \tilde{T}$
- Number of fundamental circuits: $m - n + 1$
- Performing circular rotations we can produce all the Spanning Trees.

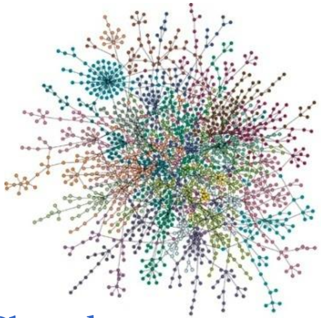
Trees



Spanning Trees

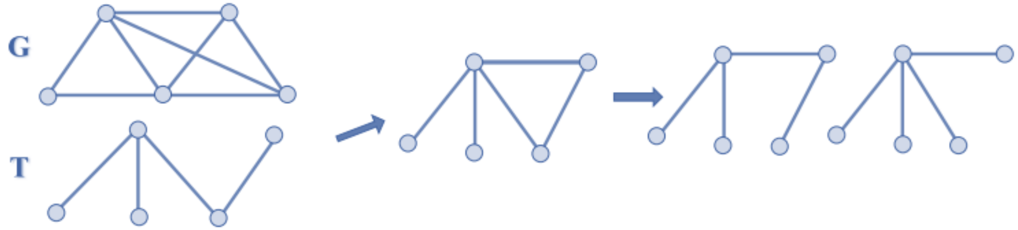
- **Fundamental Circuit**: A cycle created by a Spanning Tree and a Chord.
- Total Chords: $m - n + 1$
- $G = T \cup \tilde{T}$
- Number of fundamental circuits: $m - n + 1$
- Performing circular rotations we can produce all the Spanning Trees.
  1) Select a Spannint Tree T.
  2) Insert an edge $\Rightarrow$ C_i fundamental circuit
  3) Delete one-by-one the edges of $C_i$there are produced $T_1, T_2, \dots, T_K$ Spanning Trees
  4) Insert a new edge in C_{i+1}

Trees

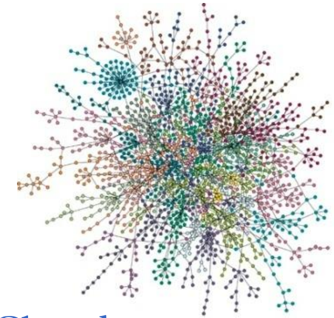


Spanning Trees

- **Fundamental Circuit**: A cycle created by a Spanning Tree and a Chord.
- Total Chords: $m - n + 1$
- $G = T \cup \tilde{T}$
- Number of fundamental circuits: $m - n + 1$
- Performing circular rotations we can produce all the Spanning Trees.
  1) Select a Spannint Tree T.
  2) Insert an edge $\Rightarrow$ C_i fundamental circuit
  3) Delete one-by-one the edges of $C_i$there are produced $T_1, T_2, \dots, T_K$ Spanning Trees
  4) Insert a new edge in C_{i+1}

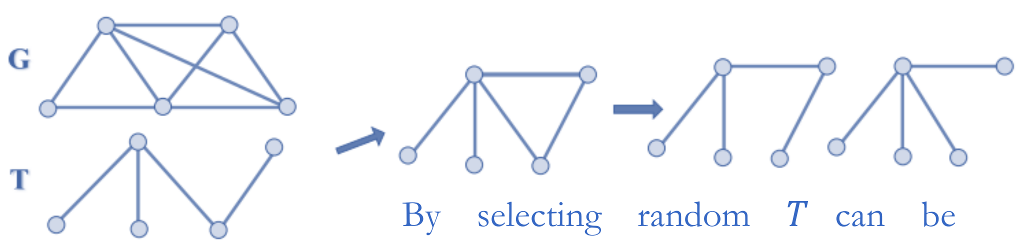


Trees



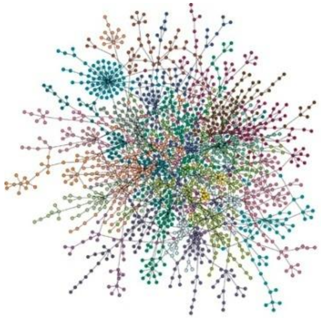
Spanning Trees

- **Fundamental Circuit**: A cycle created by a Spanning Tree and a Chord.
- Total Chords: $m - n + 1$
- $G = T \cup \tilde{T}$
- Number of fundamental circuits: $m - n + 1$
- Performing circular rotations we can produce all the Spanning Trees.
  1) Select a Spannint Tree T.
  2) Insert an edge $\Rightarrow$ C_i fundamental circuit
  3) Delete one-by-one the edges of $C_i$there are produced $T_1, T_2, \ldots, T_K$ Spanning Trees
  4) Insert a new edge in C_{i+1}



By selecting random T can be produced all the Spanning Trees of G

Trees



Spanning Trees

- **Distance** of Spanning Trees is the number of edges that belong to a Spanning Tree but not in another.
  - $dist(T_i, T_j) = dist(T_j, T_i)$
  - $dist(T_i, T_j) \geq 0, (dist(T_i, T_i) = 0)$
  - $dist(T_i, T_j) \leq dist(T_i, T_u) + dist(T_u, T_j)$

- **Theorem 15**:

  The maximum distance between two Spanning Trees T_i and T_j of a connected graph $G(V, E)$ is $\max\left(dist(T_i, T_j)\right) \leq \min(n-1, m-n-1)$

- **Central** is the Spanning Tree T_0 if it holds that $\left(dist(T_0, T_i)\right) \leq \max\left(dist(T, T_i)\right) \forall\, T$ Spanning Trees of G. Hence, a graph may has more than one Central Spanning Trees

- Weighted Spanning Trees: Find Minimum Spanning Trees utilizing the algorithms of Kruskal, Prim, or Boruvka.