Academic Year 2023/2024

Social Media Analytics (SMA)

Network and Graph Theory

Part 2 – Cliques, Clustering, Trees

Marco Viviani

University of Milano-Bicocca

Department of Informatics, Systems, and Communication

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

DIPARTIMENTO DI INFORMATICA, SISTEMISTICA E COMUNICAZIONE

Clique (*Cricca*)
Undirected graphs

- Set of vertices C totally connected in a graph G, $C \subseteq G$

- We usually ignore:
  - Single vertices
  - Vertex pairs connected by an edges (“order 2” clique)

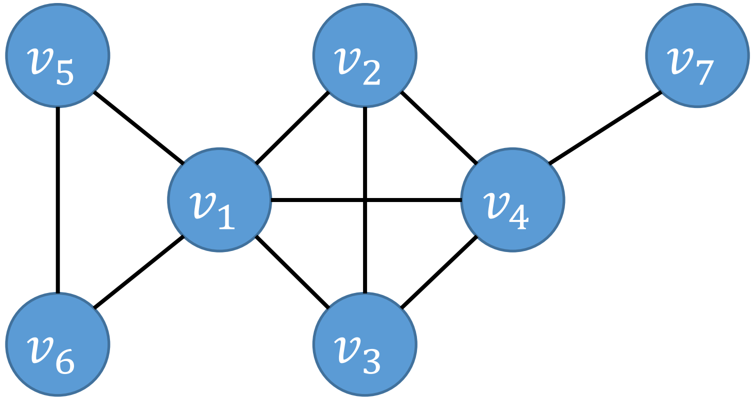
- **"Maximal" clique** (*Cricca massimale*)
  - Clique that cannot be extended by adding a new adjacent vertex that increases the size of the clique

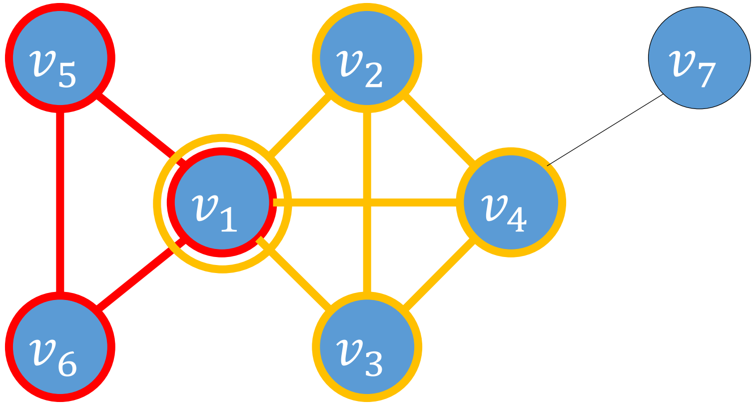
- **Maximum clique** (*Cricca massima*)
  - The largest clique in a graph G

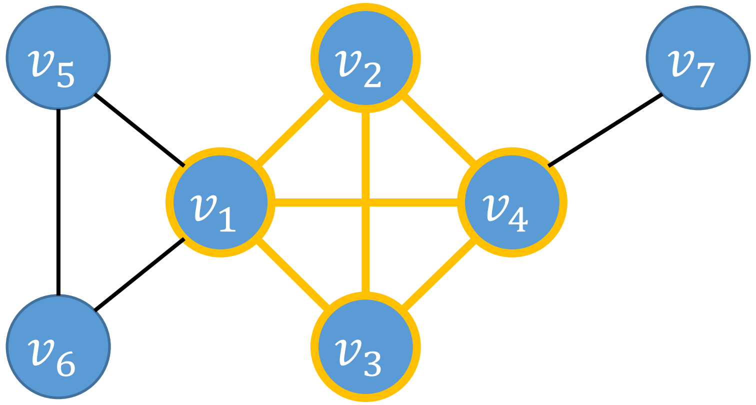
Clique

Examples on an undirected graph

- «Maximal» cliques?
  - $C_1 = \{v_1, v_5, v_6\}$
  - $C_2 = \{v_1, v_2, v_3, v_4\}$
- Maximum clique?
  - $|C_2| = 4$







Clique
Directed graphs

- In a directed graph, the concept of a **clique** is less common. Instead, you may consider a directed clique

- A **directed clique** is a subset of vertices where for every pair of distinct vertices u and v in the subset, there are two directed edges: one from u to v and another from v to u

- More formally, a directed clique in a directed graph G is a subset C of vertices such that for every pair of distinct vertices $u, v \in C$, there is a directed edge from u to v and a directed edge from v to u in G

- Similar to undirected graphs, a **maximal directed clique** is a directed clique that cannot be extended by adding an adjacent vertex

Clustering coefficient

- The **clustering** (or **aggregation**) **coefficient** is the measure of the degree to which the nodes of a graph tend to be connected to each other

- Three possibilities to calculate the clustering coefficient:
  - Local clustering coefficient
  - Average clustering coefficient
  - Global clustering coefficient

Clustering coefficient

Undirected VS directed graphs

- In an **undirected graph**, it quantifies how close a vertex's neighbors are to being a clique. It is a measure of local density

- In a **directed graph**, the concept of clustering coefficient is less straightforward
  - There are two types of clustering coefficients: in-degree clustering coefficient and out-degree clustering coefficient
  - These measures assess the likelihood that a vertex's in-neighbors and out-neighbors form cliques
  - It provides insights into the local connectivity patterns in directed graphs

Connected components and clustering coefficient

- The **connected components** divide the graph into disjoint subgraphs, and within each connected component, the **clustering coefficient** is typically high
  - This is because within a connected component, vertices are closely interconnected, and their neighbors are more likely to be connected to each other, leading to a high clustering coefficient

- In other words, connected components create a **macro-level division** of the graph into separate, densely connected regions

- The clustering coefficient does not provide information about the global connectivity or division of the graph into connected components

Strongly Connected Components (SCCs) and clustering coefficient

- SCCs and clustering coefficients address different aspects of a directed graph's structure and connectivity
  - SCCs are more concerned with global connectivity and the existence of self-contained subgraphs, while the clustering coefficient measures local clustering patterns around individual vertices
  - They can be useful in different contexts and for different analysis purposes within directed graphs

- Further details **only if needed** in the next lectures

Local clustering coefficient

Directed and undirected graphs

- Given $N(v)$ the set of neighbors of v, the **local clustering coefficient** $cc(v)$ of a vertex v is given by the number of edges between the members of $N(v)$ divided by the number of potential edges between them

Maximum number of potential edges between the vertices in $N(v)$ in a directed graph

- **Directed graph**:

$$cc(v) = \frac{||N(v)||}{k(k-1)}$$

$$k = |N(v)| = d(v)$$

- **Undirected graph**:

$$cc(v) = \frac{2||N(v)||}{k(k-1)}$$

In an undirected graph the maximum number of potential edges between the neighbors of v is $\frac{k(k-1)}{2}$

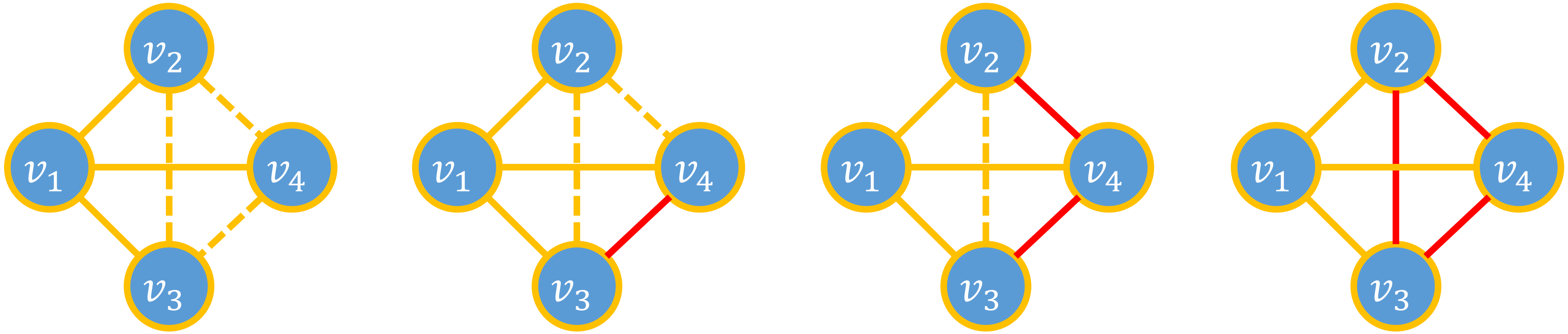
Local clustering coefficient

Examples on undirected graphs

$$cc(v) = \frac{2||N(v)||}{k(k-1)}$$

Real edge

Potential edge

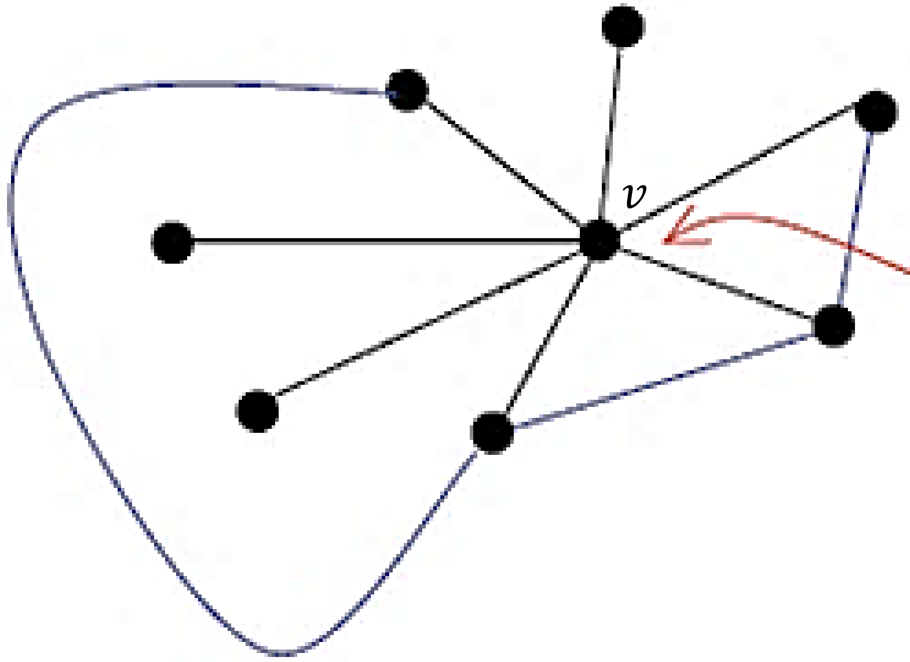


$$cc(v_1) = \frac{2*0}{3*2} = \frac{0}{6} = 0 \qquad cc(v_1) = \frac{2*1}{3*2} = \frac{1}{3} \qquad cc(v_1) = \frac{2}{3} \qquad cc(v_1) = \frac{6}{6} = 1$$

Local clustering coefficient

Exercise on undirected graphs

- Calculate the local clustering coefficient of node v in the following graph:



$$cc(v) = \frac{2||N(v)||}{k(k-1)} = ?$$

$$cc(v) = \frac{2 * 3}{7 * 6} = \frac{6}{42} = \frac{1}{7} = 0{,}14$$

Average clustering coefficient

Directed and undirected graphs

- The **average clustering coefficient** $cc(G)$ of a graph G is given by the average of the clustering coefficients for each single node of the graph

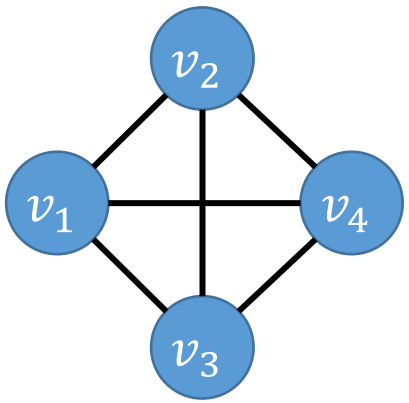
- Formally:

$$cc(G) = \frac{1}{|V|}\sum_{i=1}^{n} cc(v_i)$$

Average clustering coefficient
Examples on undirected graphs

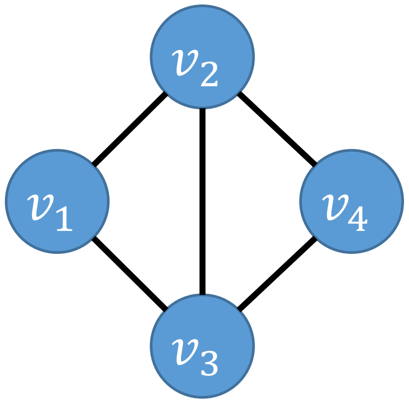
- $cc(G_1) = \frac{1}{4}(1 + 1 + 1 + 1) = 1$

- $cc(G_2) = \frac{1}{4}\left(1 + \frac{2}{3} + \frac{2}{3} + 1\right) = \frac{5}{6} = 0{,}8\overline{3}$
    - $cc(v_1) = \frac{2*1}{2*1} = 1$
    - $cc(v_2) = \frac{2*2}{3*2} = 2/3$
    - $cc(v_3) = \frac{2*2}{3*2} = 2/3$
    - $cc(v_4) = \frac{2*1}{2*1} = 1$



G_1

$$cc(G) = \frac{1}{|V|}\sum_{i=1}^{n} cc(v_i)$$



G_2

Global clustering coefficient
Directed and undirected graphs

- The concept of **global clustering coefficient** (a.k.a. **transitivity**) is based on triples (triads) of vertices.
  - Open triplet: three nodes connected by two edges
  - Closed triplet: three nodes connected by three edges

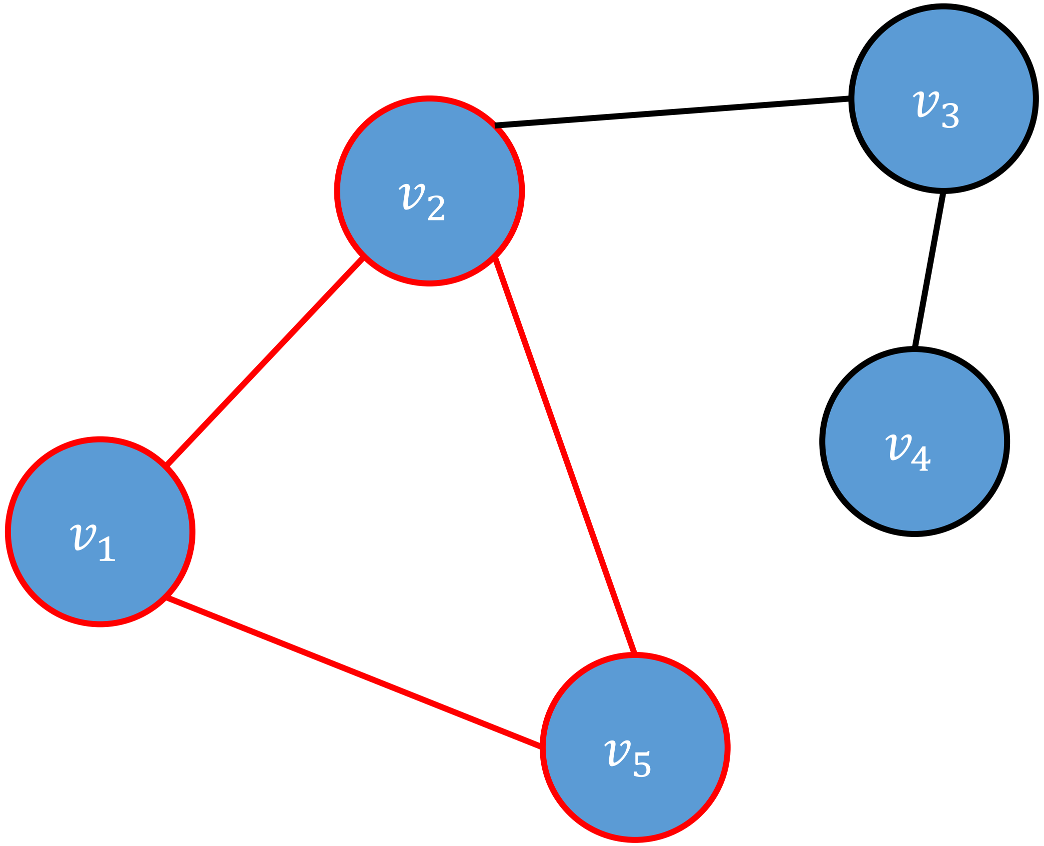
- Each triple is **centered** around a vertex

- A **triangle** consists of three closed triples centered on the same three nodes that compose them

Triangle

Example on undirected graphs

Vertex	Triplets centered around the vertex
v_1	$\langle v_1, v_2, v_5 \rangle$
v_2	$\langle v_1, v_2, v_3 \rangle$ $\langle v_1, v_2, v_5 \rangle$ $\langle v_2, v_3, v_5 \rangle$
v_3	$\langle v_2, v_3, v_4 \rangle$
v_4	—
v_5	$\langle v_1, v_2, v_5 \rangle$



Global clustering coefficient

Formal definition for undirected graphs

- The **global clustering coefficient** $cc_{\triangle}(G)$ of a graph G is calculated as the number of closed triples (or 3 times the number of triangles) divided by the total number of triples (open and closed ones)

- Formally:

Number of triangles in the graph

$$cc_{\triangle}(G) = \frac{3 * n_{\triangle}(G)}{n_{\wedge}(G)} = \frac{\sum_{i=1}^{n}(cc(v_i) * \omega_i)}{\sum_{i=1}^{n} \omega_i}$$

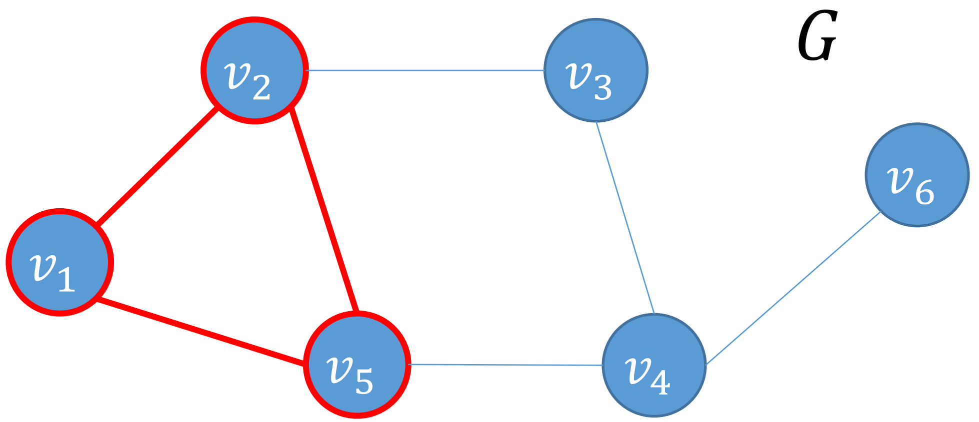
Total number of triples (open and closed) in the graph

Number of triples in which the node v_i is central («weight» of the node v_i)

Global clustering coefficient

Example 1 on undirected graphs

Vertex	Triplets centered around the vertex	Weight (ω_i)
v_1	$\langle v_1, v_2, v_5 \rangle$	1
v_2	$\langle v_1, v_2, v_3 \rangle$ $\langle v_1, v_2, v_5 \rangle$ $\langle v_2, v_3, v_5 \rangle$	3
v_3	$\langle v_2, v_3, v_4 \rangle$	1
v_4	$\langle v_3, v_4, v_5 \rangle$ $\langle v_3, v_4, v_6 \rangle$ $\langle v_4, v_5, v_6 \rangle$	3
v_5	$\langle v_1, v_2, v_5 \rangle$ $\langle v_1, v_4, v_5 \rangle$ $\langle v_2, v_4, v_5 \rangle$	3
v_6	—	0

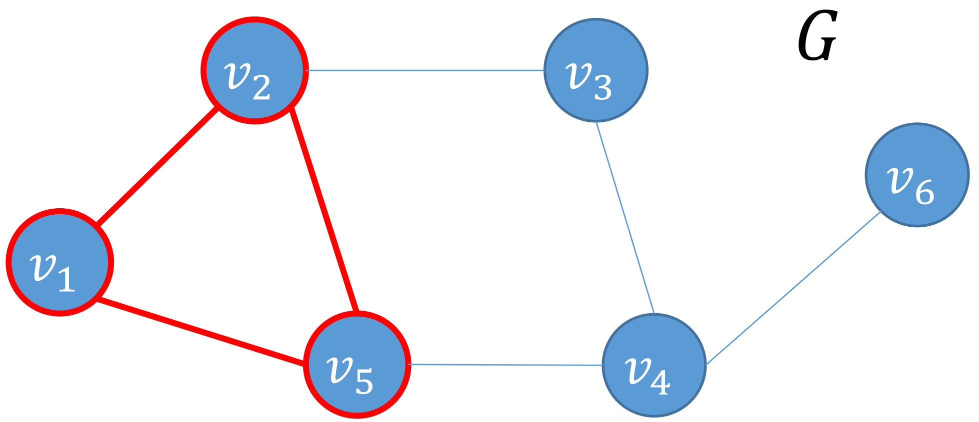


$$cc_{\triangle}(G) = \frac{3 * n_{\triangle}(G)}{n_{\wedge}(G)} = \frac{3 * 1}{11} = \frac{3}{11}$$

Global clustering coefficient

Example 2 on undirected graphs

Vertex	Weight (ω_i)	$cc(v_i)$
v_1	1	$2 * 1/2 * 1 = 1$
v_2	3	$2 * 1/3 * 2 \; = \; 1/3$
v_3	1	$2 * 0/2 * 1 \; = \; 0$
v_4	3	$2 * 0/3 * 2 = 0$
v_5	3	$2 * 1/3 * 2 = 1/3$
v_6	0	0



$$cc(v) = \frac{2||N(v)||}{k(k-1)}$$

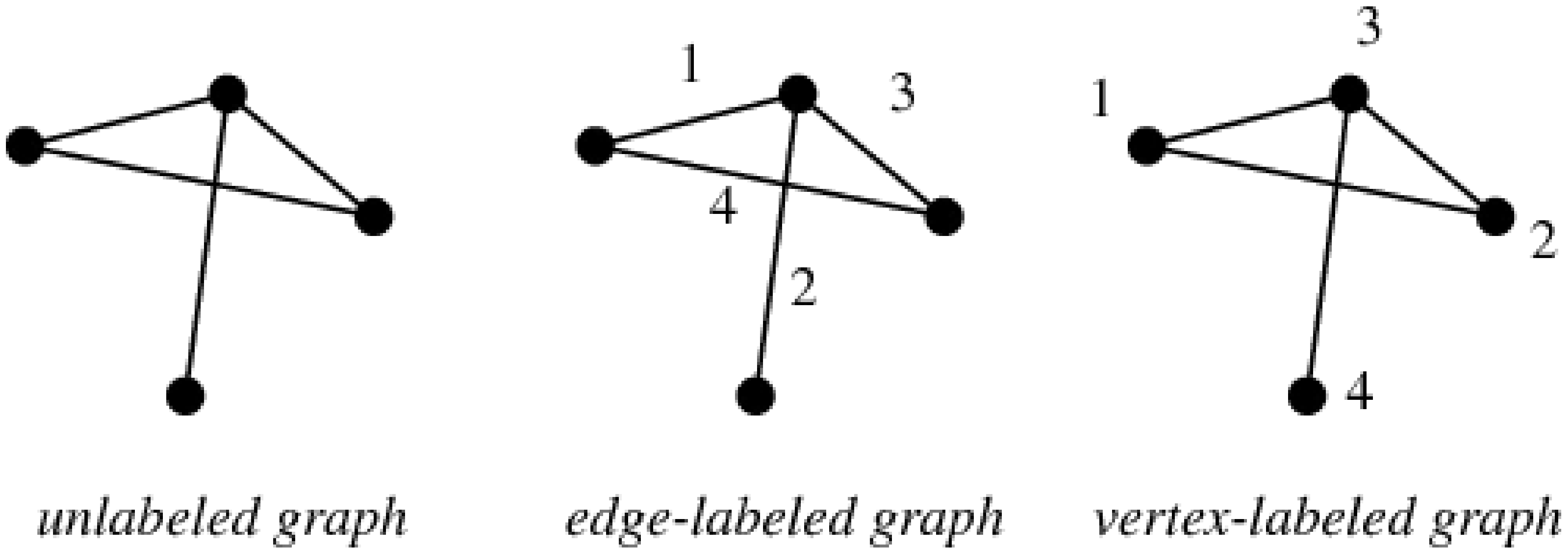
$$cc_{\triangle}(G) = \frac{\sum_{i=1}^{n}(cc(v_i) * \omega_i)}{\sum_{i=1}^{n}\omega_i} = \frac{(1 * 1) + \left(\frac{1}{3} * 3\right) + (0 * 1) + (0 * 3) + \left(\frac{1}{3} * 3\right) + 0}{11} = \frac{3}{11}$$

Labeled graphs and weighted graphs

Grafi etichettati e grafi pesati

- A **labeled graph** (directed or undirected) is a graph in which an additional information called a label is associated with each arc or vertex
- A **weighted graph** is (generally) a graph labeled on edges with non-negative numbers called weights
- Given a path, the **total weight** of the path is (generally) the sum of the weights on the edges in the path
  - Application example: map with roads (also one-way) labeled by the distances between cities

Examples of labeled graphs

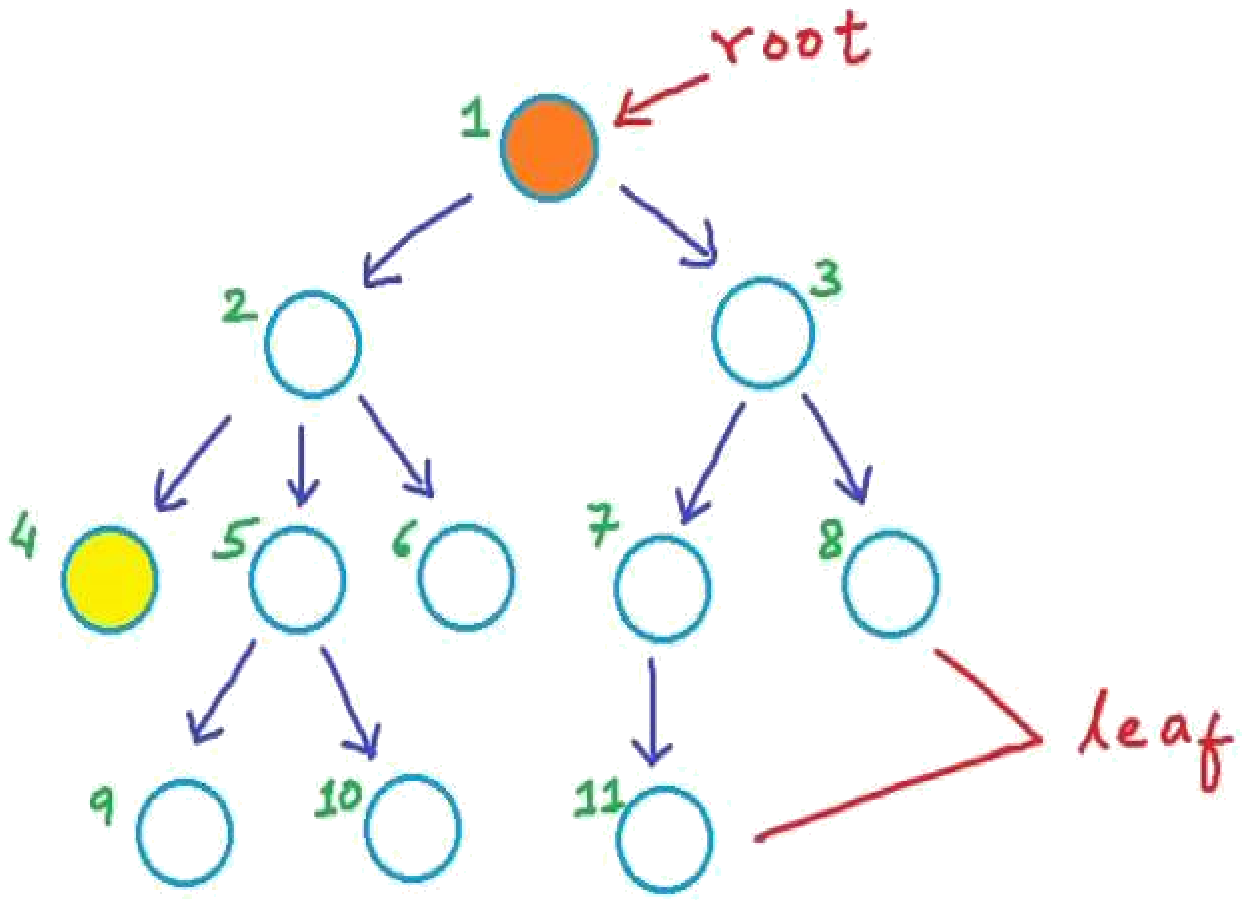


Definition of tree

- An **undirected tree** is an undirected, connected, and acyclic graph in which a node is designated as the root

- A **directed tree** is a directed graph that is empty or has a root node such that:
  - There are no arcs entering the root
  - Each non-root node has exactly one incoming edge
  - For each non-root node there is a path that goes from the root to the node itself

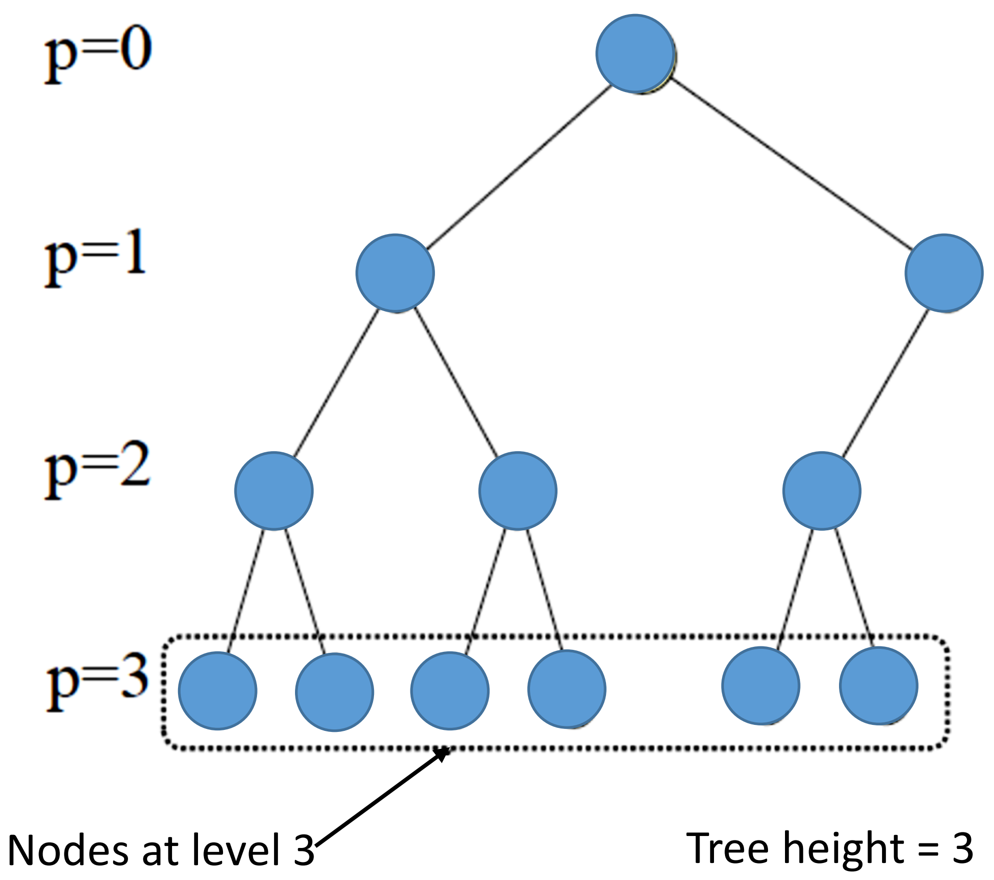
Introduction to Trees



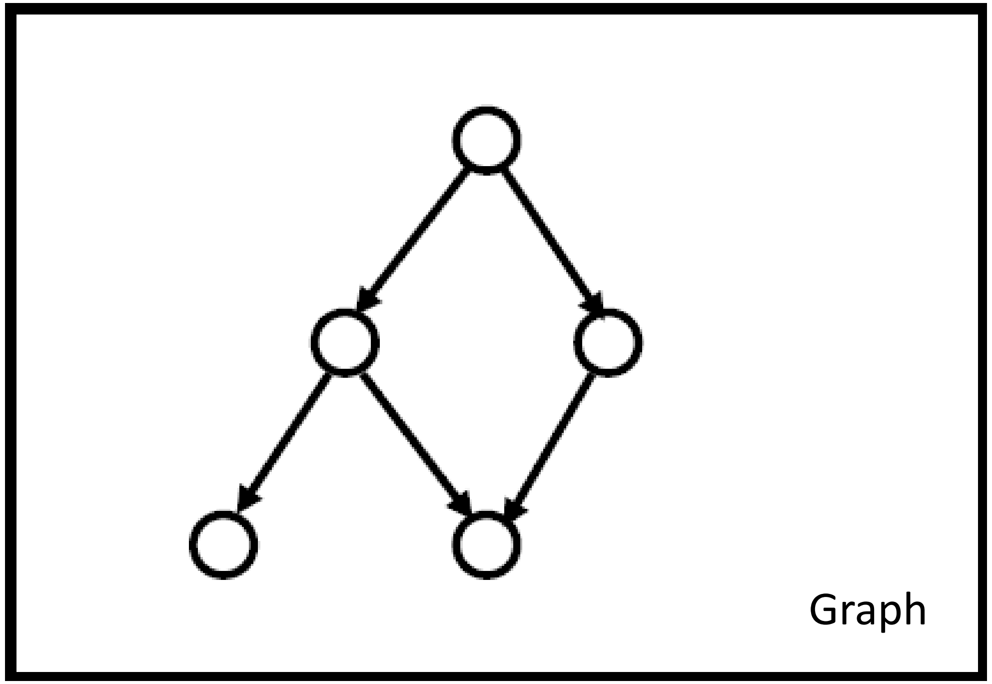
root
children
Parent
Sibling → have same parent
leaf → has no child

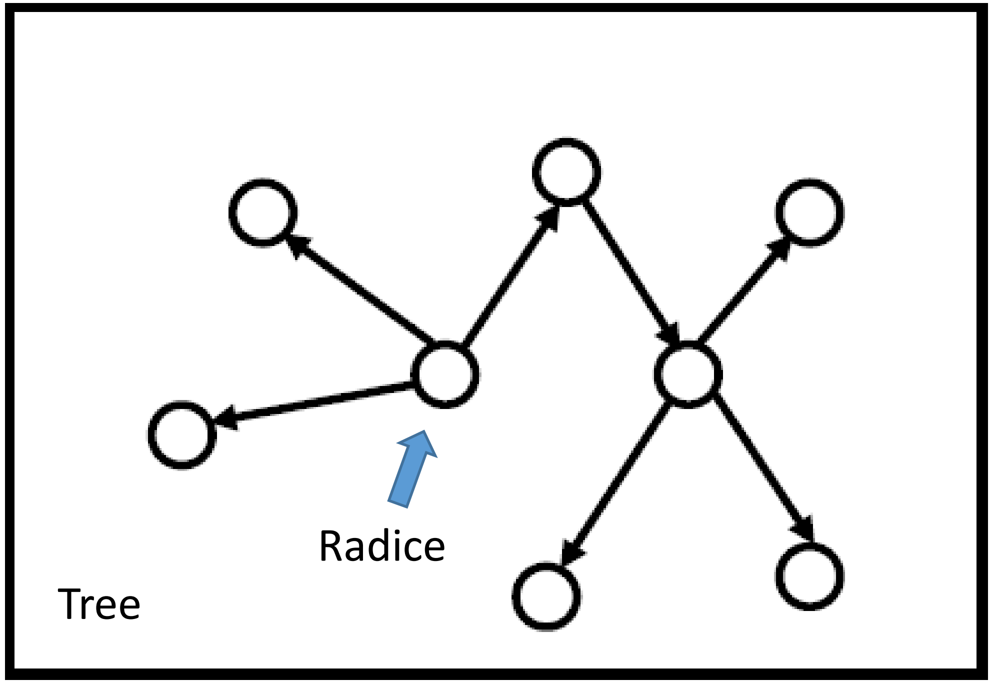
Other definitions

- In a tree
  - Depth of a node: the length of the path from the root to the node (i.e., number of edges crossed)
  - Level: the set of nodes at the same depth
  - Tree height: maximum depth reached by the leaves

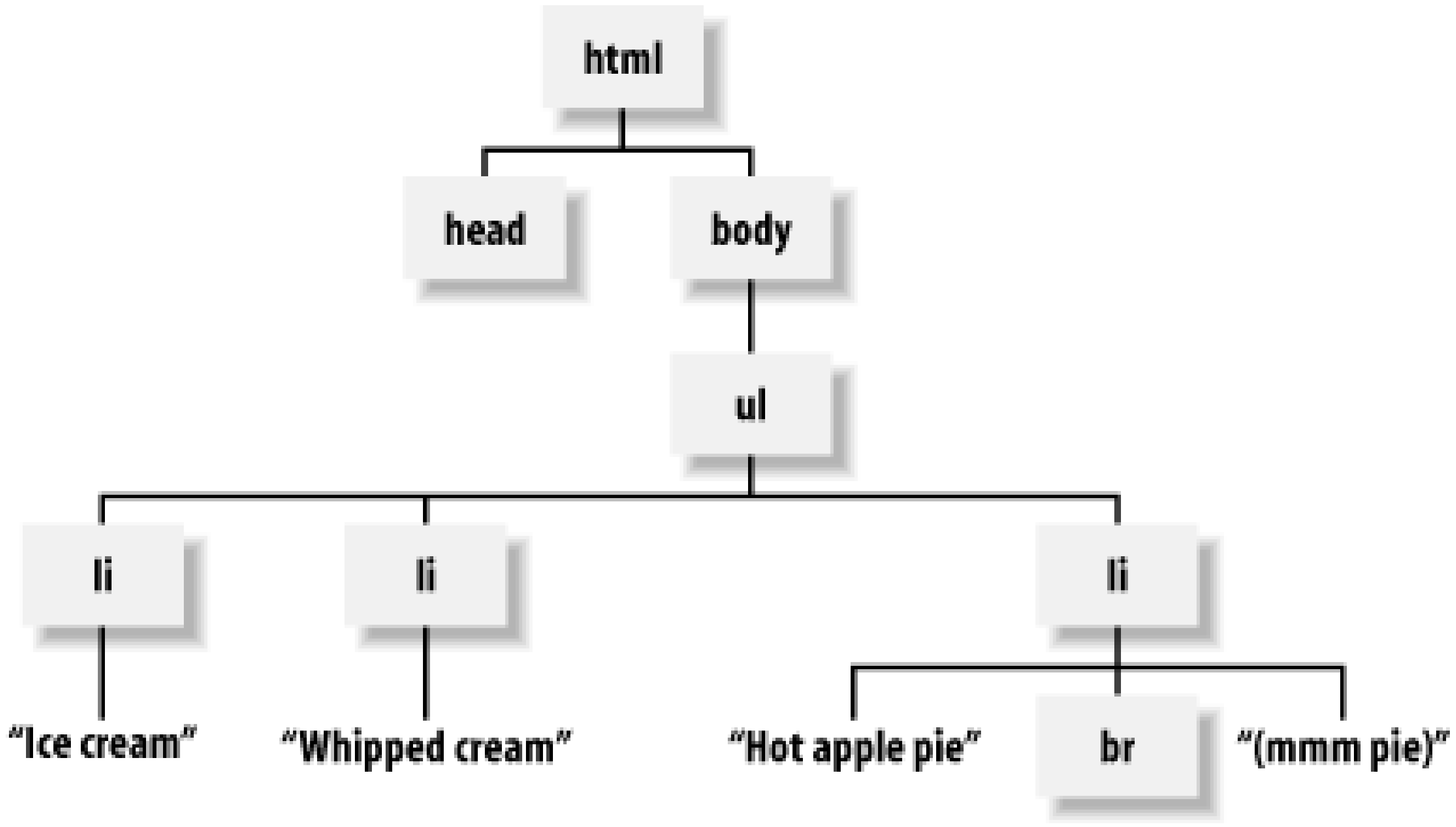


Are there any trees?



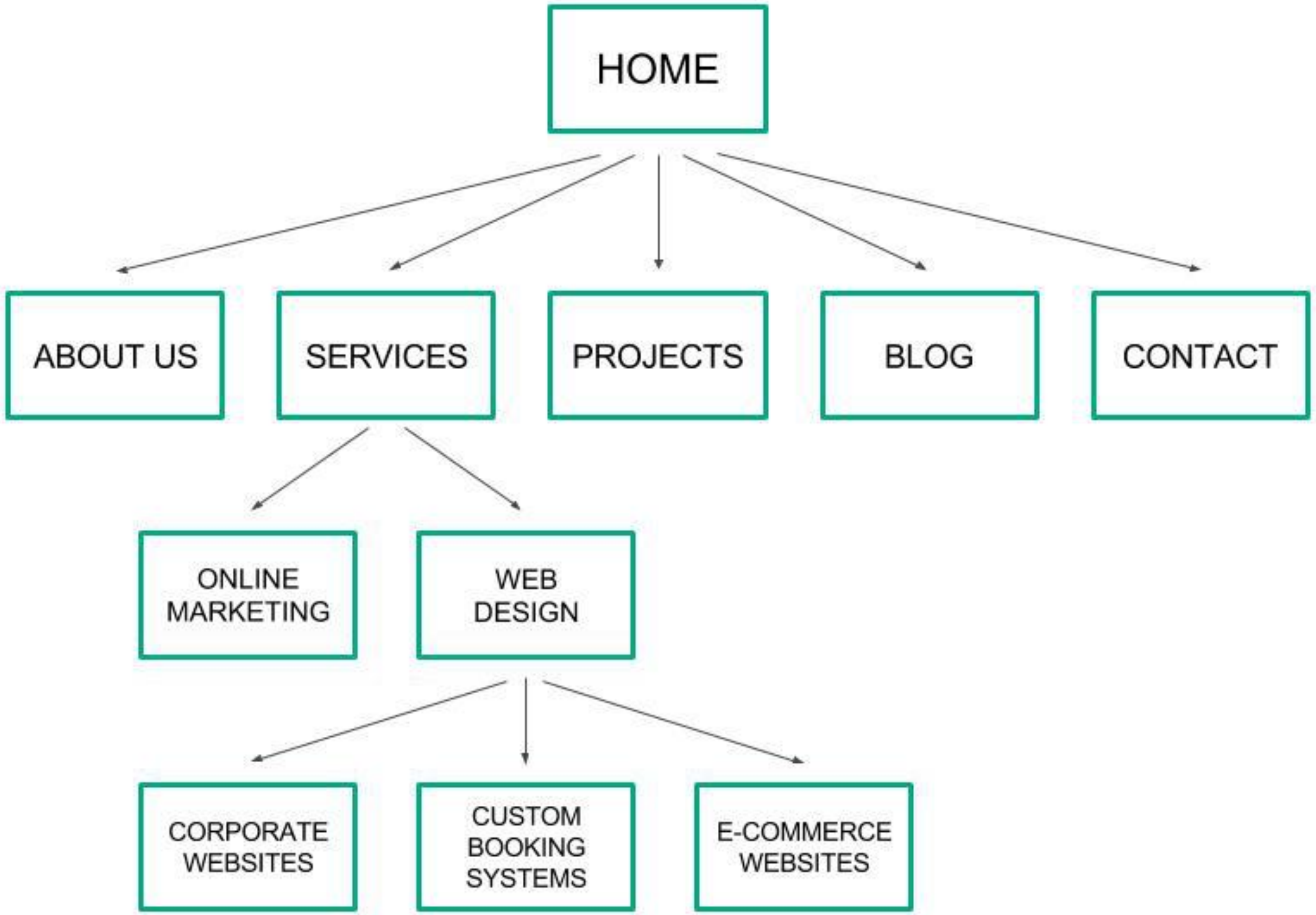


HTML tree

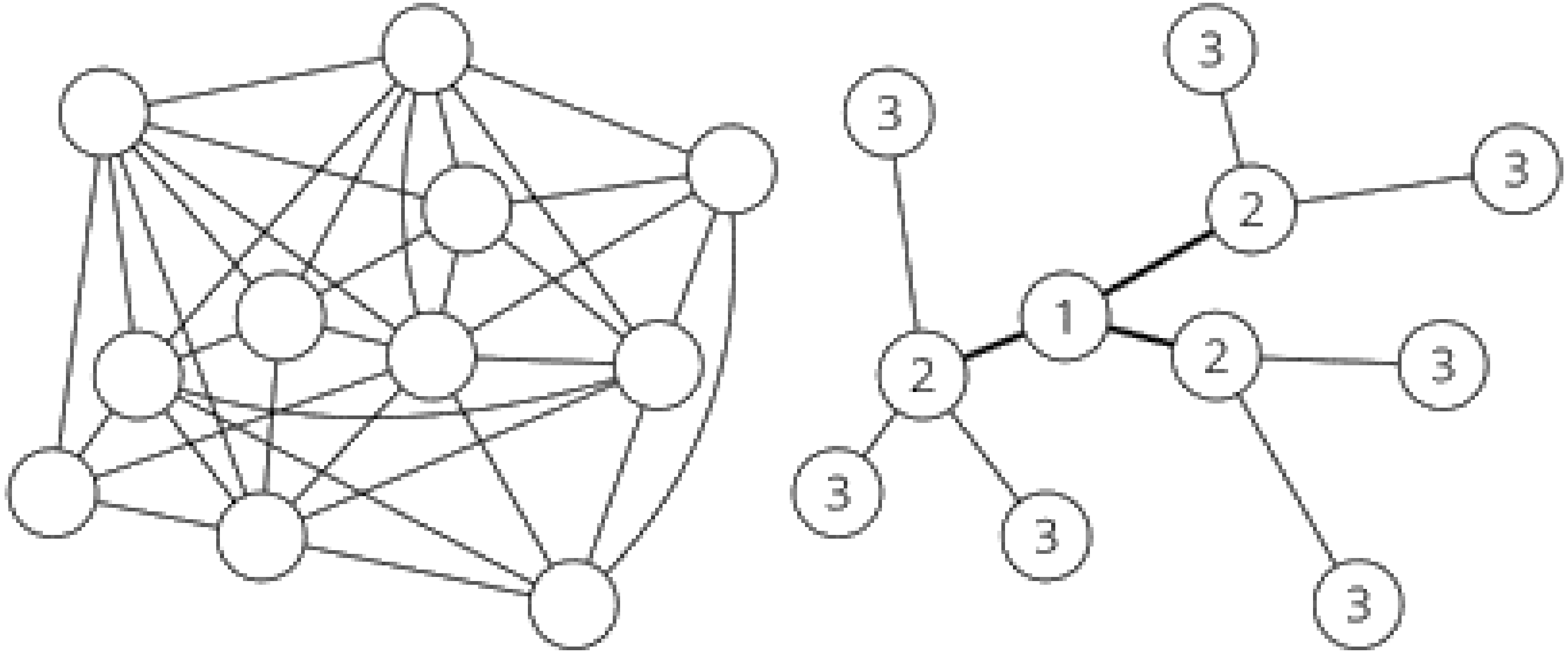


html
head
body
ul
li
li
li
"Ice cream"
"Whipped cream"
"Hot apple pie"
br
"(mmm pie)"

Website structure tree



Social media and trees



https://intersectproject.org/faith-and-culture/why-its-so-hard-to-fix-social-media/

Tree traversal

- **Tree traversal** refers to the process of visiting (searching and/or updating) each node in a tree structure exactly once
- Traversal algorithms are classified according to the order in which nodes are visited

Algorithms for tree traversal

- **Depth-First Search** (DFS)
  - Branches are visited, one after the other
  - Three variations

- **Breadth-First Search** (BFS)
  - In layers, starting from the root

Graph traversal

- **Graph traversal** refers to the process of visiting (searching and/or updating) each vertex in a graph
- Traversal algorithms are classified according to the **order** in which nodes are visited
- Tree traversal is a special case of graph traversal