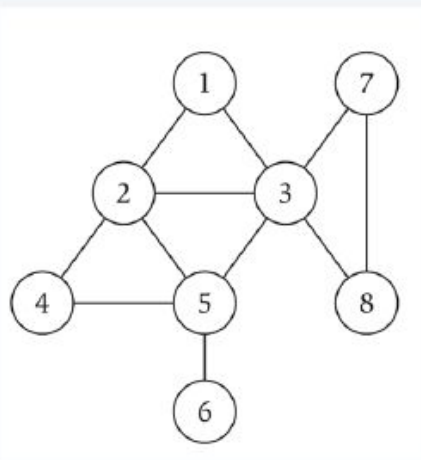


Algorithms: Basic Graph Terminologies and Definitions

Undirected graphs

Notation. $G = (V, E)$

- V = nodes (or vertices).
- E = edges (or arcs) between pairs of nodes.
- Captures pairwise relationship between objects.
- Graph size parameters: $n = |V|, m = |E|$.



$$V = \{ 1, 2, 3, 4, 5, 6, 7, 8 \}$$

$$E = \{ 1-2, 1-3, 2-3, 2-4, 2-5, 3-5, 3-7, 3-8, 4-5, 5-6, 7-8 \}$$

$$m = 11, n = 8$$

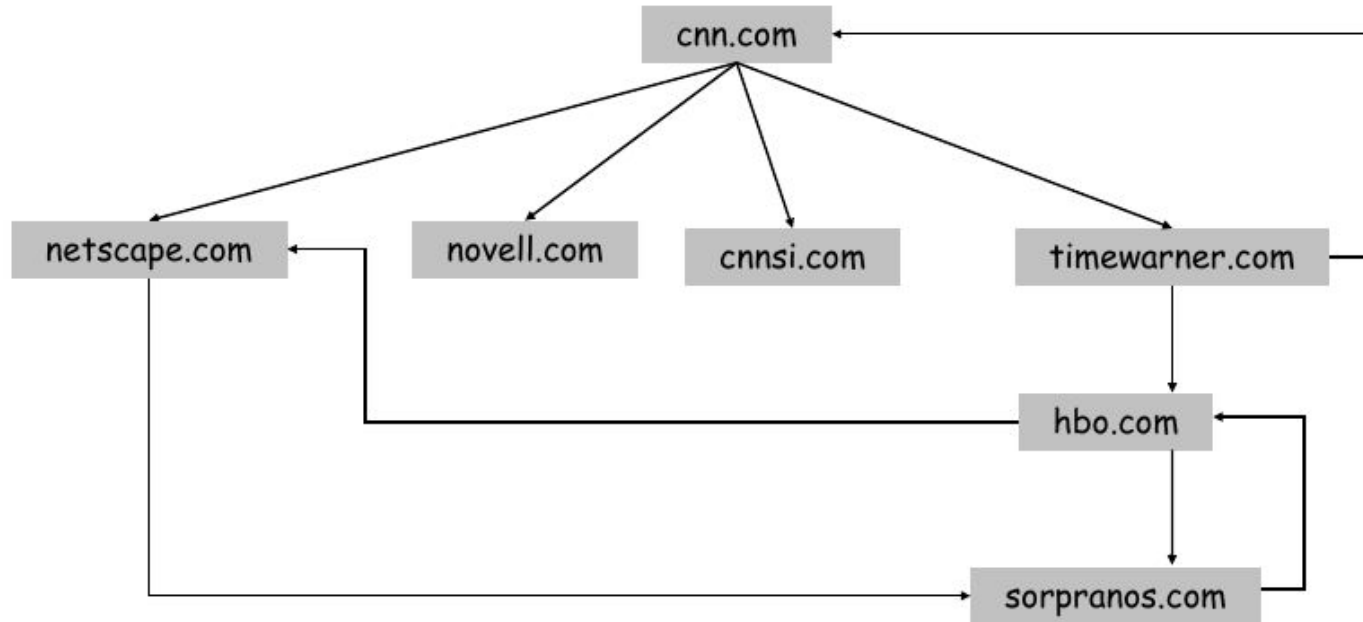
Some graph applications

graph	node	edge
communication	telephone, computer	fiber optic cable
circuit	gate, register, processor	wire
mechanical	joint	rod, beam, spring
financial	stock, currency	transactions
transportation	street intersection, airport	highway, airway route
internet	class C network	connection
game	board position	legal move
social relationship	person, actor	friendship, movie cast
neural network	neuron	synapse
protein network	protein	protein-protein interaction
molecule	atom	bond

World Wide Web

Web graph.

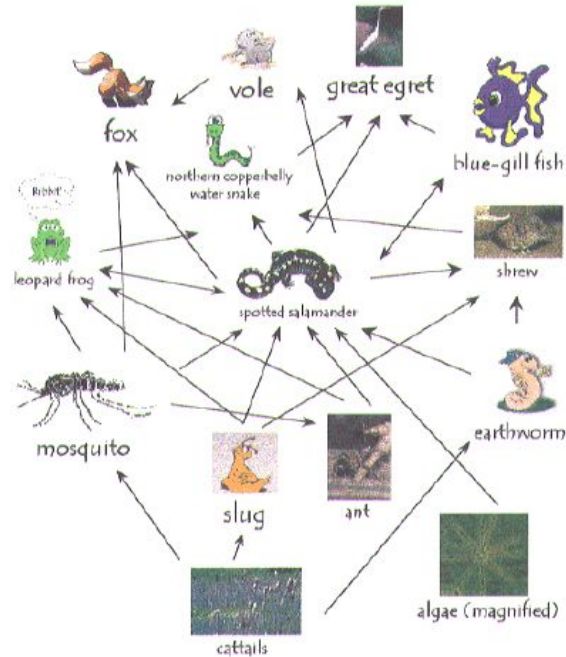
- Node: web page.
- Edge: hyperlink from one page to another.



Ecological Food Web

Food web graph.

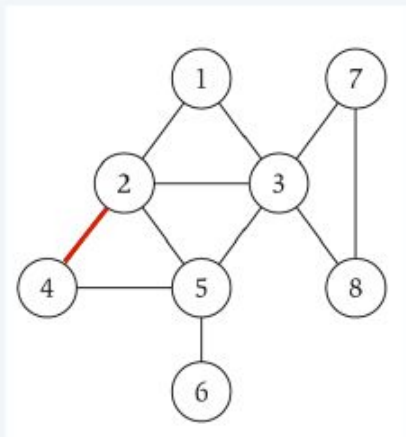
- Node = species.
- Edge = from prey to predator.



Graph representation: adjacency matrix

Adjacency matrix. n -by- n matrix with $A_{uv} = 1$ if (u, v) is an edge.

- Two representations of each edge.
- Space proportional to n^2 .
- Checking if (u, v) is an edge takes $\Theta(1)$ time.
- Identifying all edges takes $\Theta(n^2)$ time.



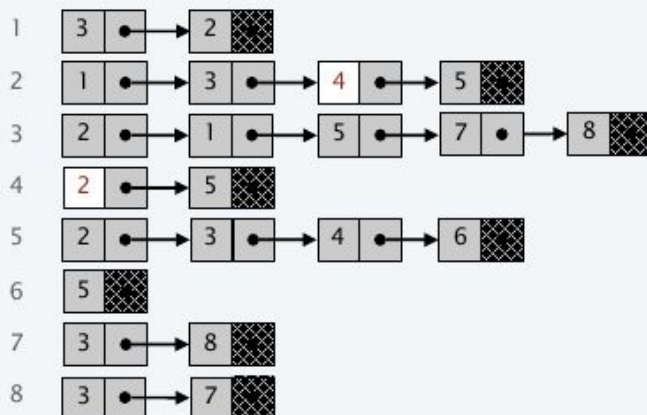
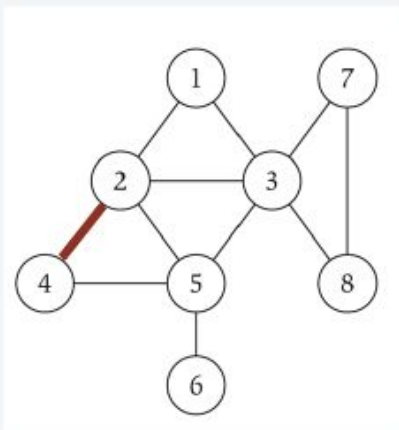
	1	2	3	4	5	6	7	8
1	0	1	1	0	0	0	0	0
2	1	0	1	1	1	0	0	0
3	1	1	0	0	1	0	1	1
4	0	1	0	0	1	0	0	0
5	0	1	1	1	0	1	0	0
6	0	0	0	0	1	0	0	0
7	0	0	1	0	0	0	0	1
8	0	0	1	0	0	0	1	0

Graph representation: adjacency lists

Adjacency lists. Node-indexed array of lists.

- Two representations of each edge.
- Space is $\Theta(m + n)$.
- Checking if (u, v) is an edge takes $O(\text{degree}(u))$ time.
- Identifying all edges takes $\Theta(m + n)$ time.

degree = number of neighbors of u

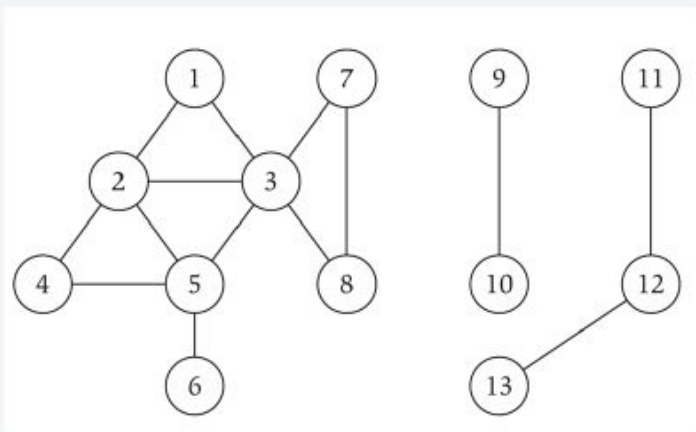


Paths and connectivity

Def. A **path** in an undirected graph $G = (V, E)$ is a sequence of nodes v_1, v_2, \dots, v_k with the property that each consecutive pair v_{i-1}, v_i is joined by a different edge in E .

Def. A path is **simple** if all nodes are distinct.

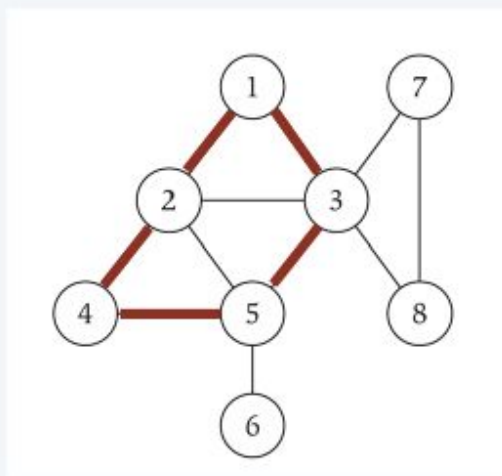
Def. An undirected graph is **connected** if for every pair of nodes u and v , there is a path between u and v .



Cycles

Def. A **cycle** is a path v_1, v_2, \dots, v_k in which $v_1 = v_k$ and $k \geq 2$.

Def. A cycle is **simple** if all nodes are distinct (except for v_1 and v_k).



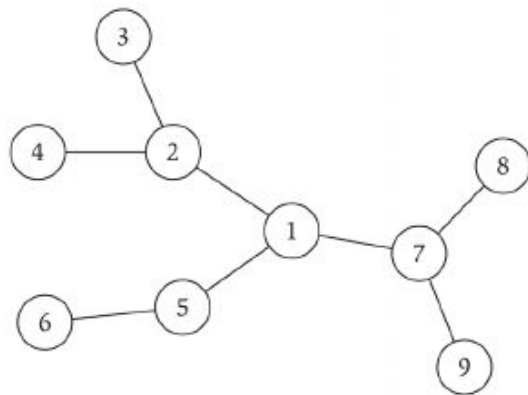
cycle C = 1-2-4-5-3-1

Trees

Def. An undirected graph is a **tree** if it is connected and does not contain a cycle.

Theorem. Let G be an undirected graph on n nodes. Any two of the following statements imply the third:

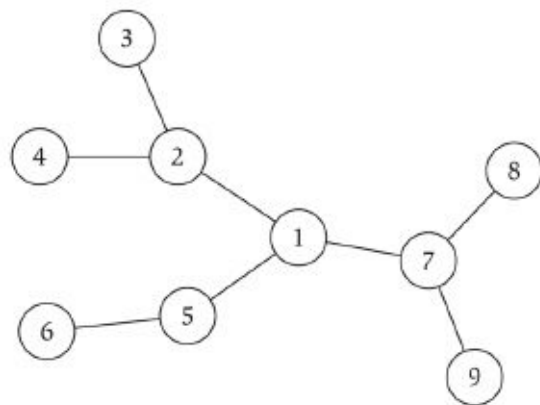
- G is connected.
- G does not contain a cycle.
- G has $n - 1$ edges.



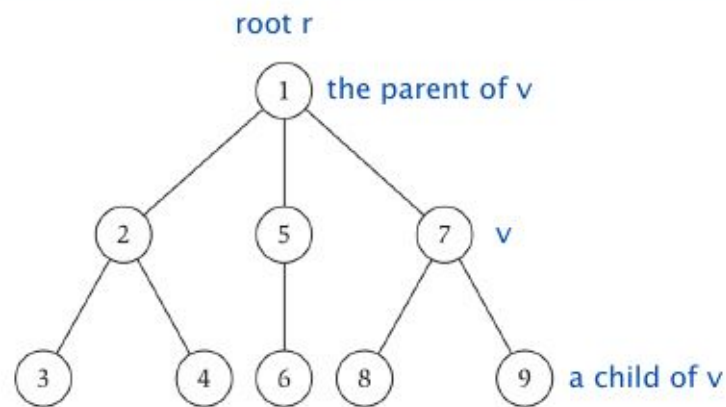
Rooted trees

Given a tree T , choose a root node r and orient each edge away from r .

Importance. Models hierarchical structure.



a tree

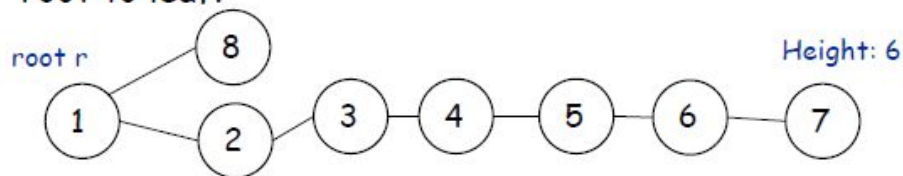


the same tree, rooted at 1

Binary Tree

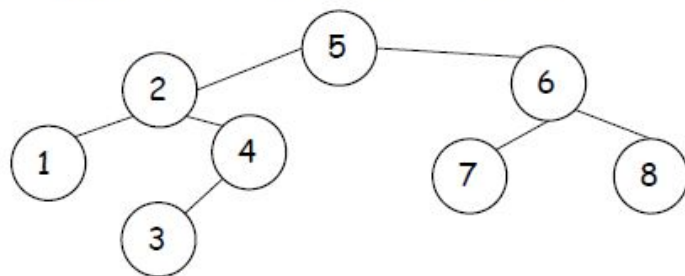
Def. A rooted tree in which each node has at most 2 children

Def. Height of a tree is the number of edges in the longest path from root to leaf.



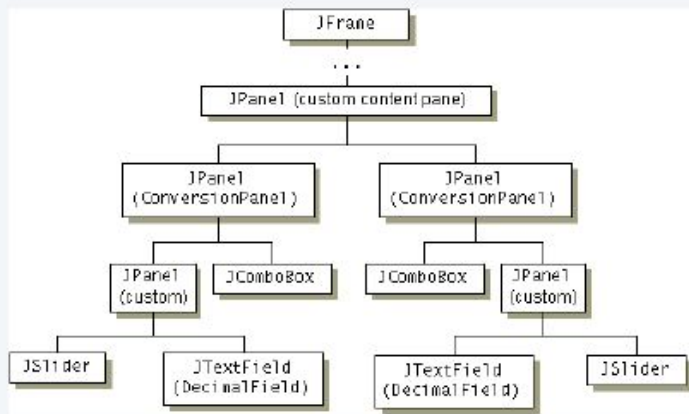
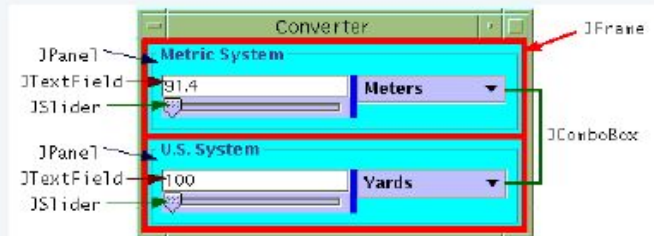
Thm. Number of nodes in binary tree of height h is $n \leq 2^{h+1} - 1$ ($= 2^0 + 2^1 + 2^2 + \dots + 2^h$).

Balanced Binary Tree. Height $h = O(\log n)$



GUI containment hierarchy

Describe organization of GUI widgets.



Suggested Reading

- Algorithm Design by Jon Kleinberg, Eva Tardos
 - ◆ Chapter 3
 - Section: 3.1

