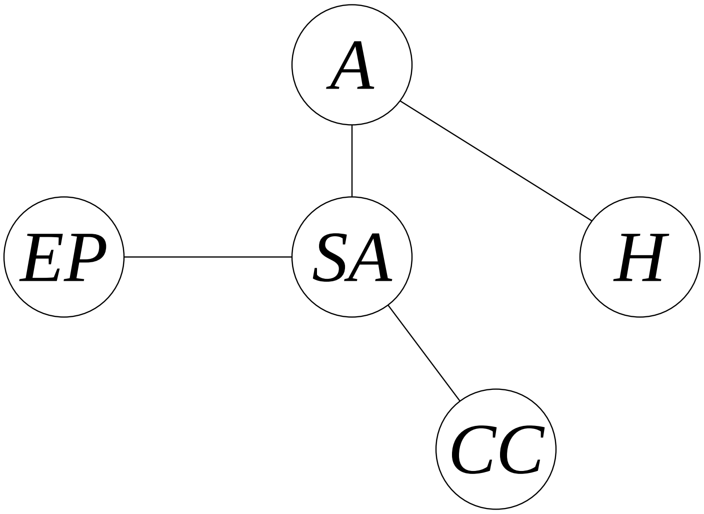
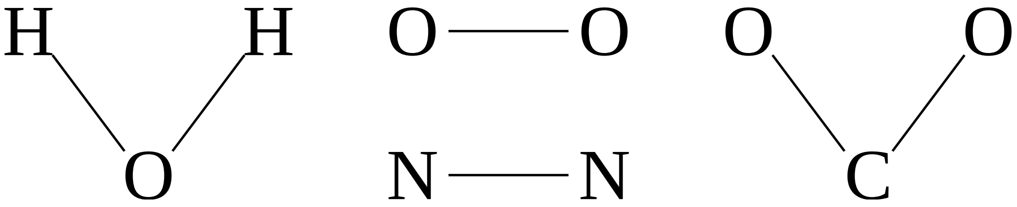
Trees

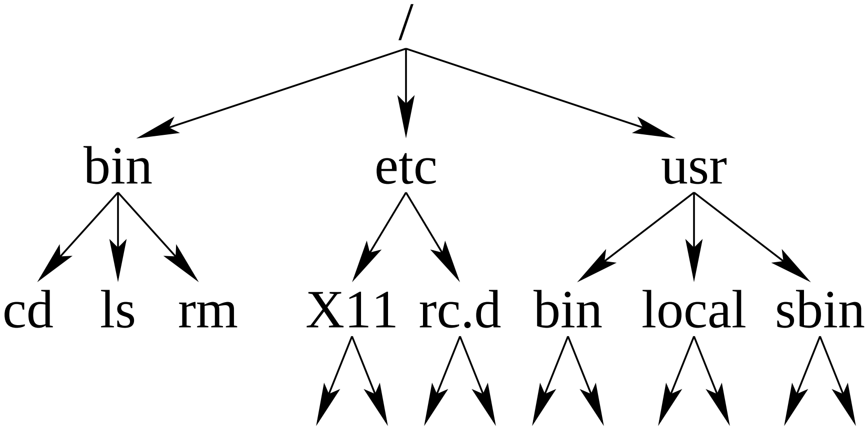
A *tree* is a connected undirected graph with no simple circuits.



A *forest* is a undirected graph with no simple circuits. A forest is a set of trees.



In a *rooted tree*, one vertex is the *root*, and all edges are directed away from the root.



Rooted Tree Terminology

If there is an edge (u, v) in a rooted tree, u is the *parent* of v, and v is a *child* of u.

u and v are *siblings* if they have same parent.

If there is a path from u to v in a rooted tree, u is an *ancestor* of v, v is a *descendent* of v, and v is in u's *subtree*.

The *height* of a tree is the length of the longest path from the root to a vertex.

If u has children, then u is *internal*.
If u has no children, then u is a *leaf*.

A rooted tree is an *m-ary* tree if every internal vertex has $\leq m$ children. It is a *full* m-ary tree if every internal vertex has exactly m children. A *binary* tree is an m-ary tree with $m = 2$.

In an *ordered rooted tree*, siblings are ordered. In an ordered binary tree, an internal vertex has a *left* child and/or a *right* child.

Properties of Trees

A tree with n vertices has $n-1$ edges.
Proof: Every vertex except the root has an edge from its parent.

A full m-ary tree with i internal vertices has $n = mi + 1$ vertices.
Proof: Each internal vertex has m edges to its children. mi edges imply $mi + 1$ vertices.

A binary tree of height h has up to 2^h leaves.
Proof later.

A binary tree of height h has $\leq 2^{h+1} - 1$ vertices.
Proof: Solve for $n = i + l$, $n = 2i + 1$, and $l = 2^h$.

The minimum height of a binary tree with l leaves is $\lceil \log l \rceil$.

The minimum height of a binary tree with n vertices is $\lfloor \log n \rfloor$.

Number of Leaves of a Binary Tree

Predicate $P(h)$:
A binary tree of height h has $\leq 2^h$ leaves.

Basis $P(0)$:
A binary tree of height 0 has $2^0 = 1$ leaf.

Induction: Prove $P(k) \rightarrow P(k+1)$

Assume $P(k)$.
A binary tree of height k has $\leq 2^k$ leaves.

Show $P(k+1)$:
A binary tree of height $k+1$ has $\leq 2^{k+1}$ leaves.

Proof:
The root of a binary tree of height $k+1$ can have 2 subtrees of height k (or less).

By the inductive assumption, a binary tree of height k has $\leq 2^k$ leaves.

Therefore, 2 subtrees of height k have $\leq 2(2^k) = 2^{k+1}$ leaves.

Tree Traversal

```
procedure preorder($T$: ordered rooted tree)
    process root of $T$
    for each subtree $S$ of $T$ from left to right
        preorder($S$)
end procedure
```

```
procedure postorder($T$: ordered rooted tree)
    for each subtree $S$ of $T$ from left to right
        postorder($S$)
    process root of $T$
end procedure
```

```
procedure inorder($T$: ordered rooted tree)
    if $T$ has subtrees
        then inorder(first subtree of $T$)
    process root of $T$
    if $T$ has subtrees
        then for remaining subtrees $S$ of $T$
                 inorder($S$)
end procedure
```