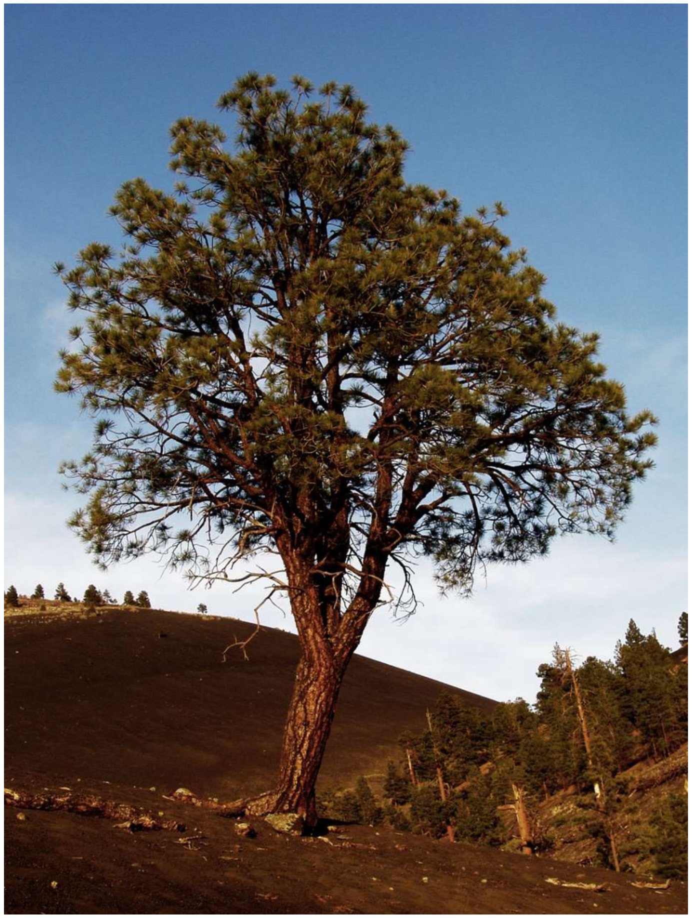
Trees

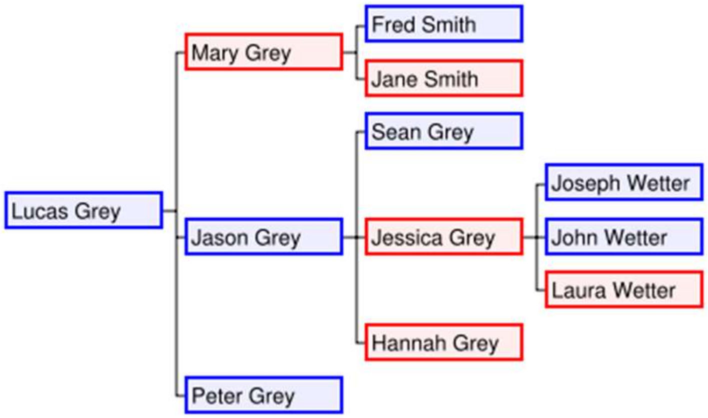
What is a Tree?

- A plant



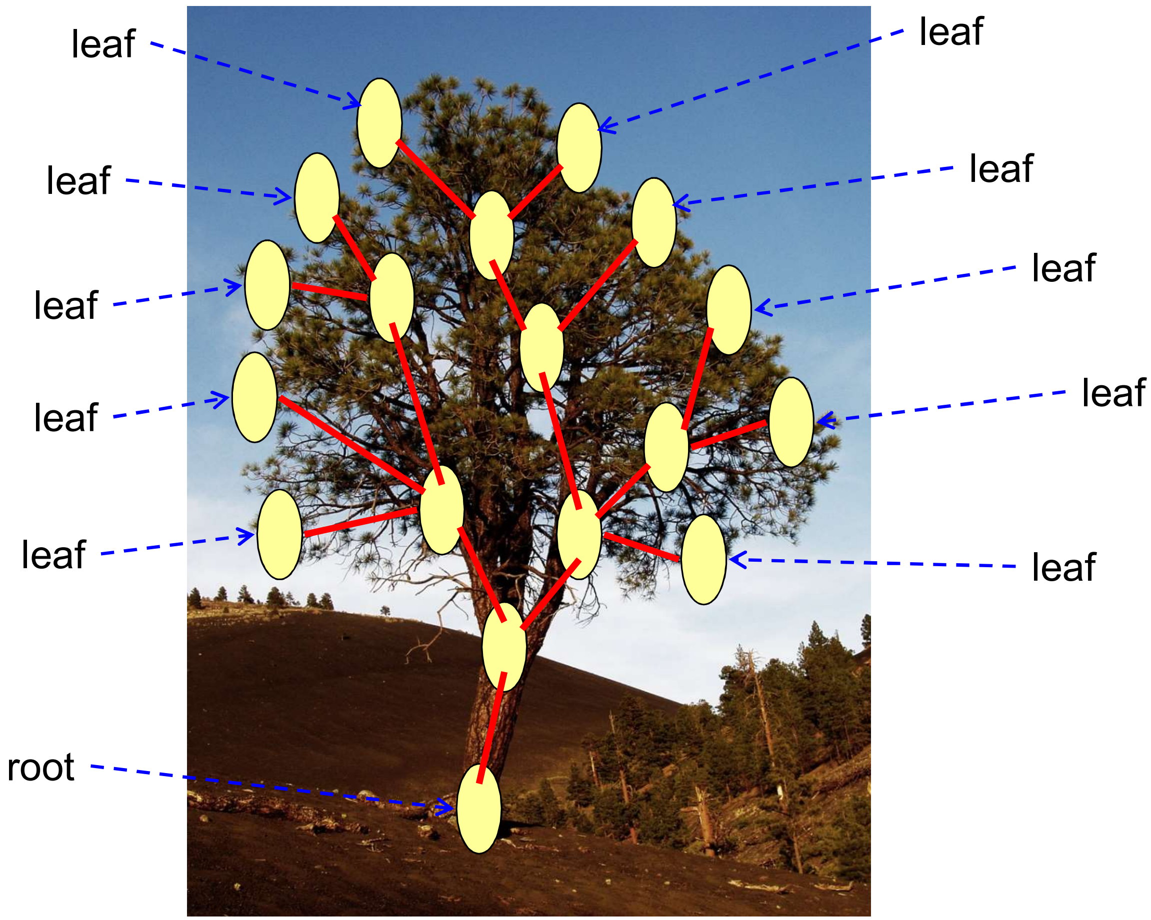
What is a Tree?

- Family tree

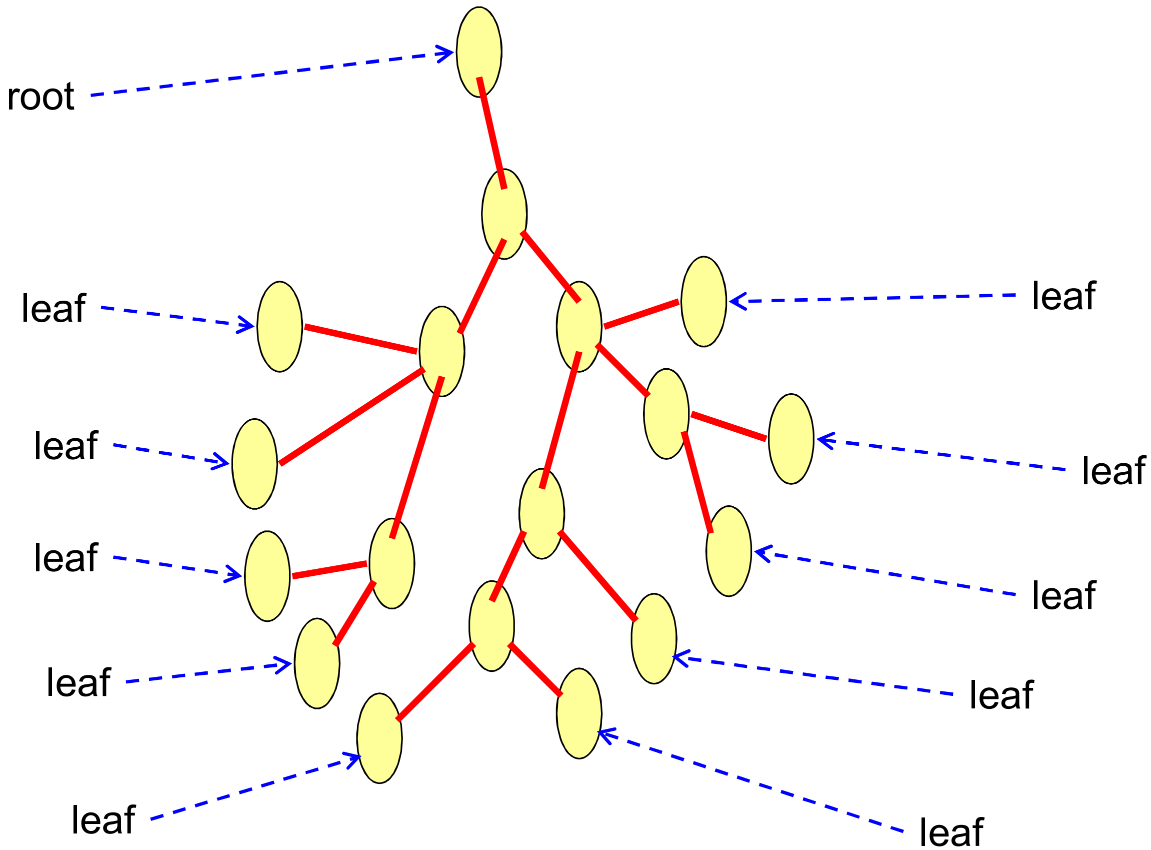


Source: Wikipedia

What is a Tree?



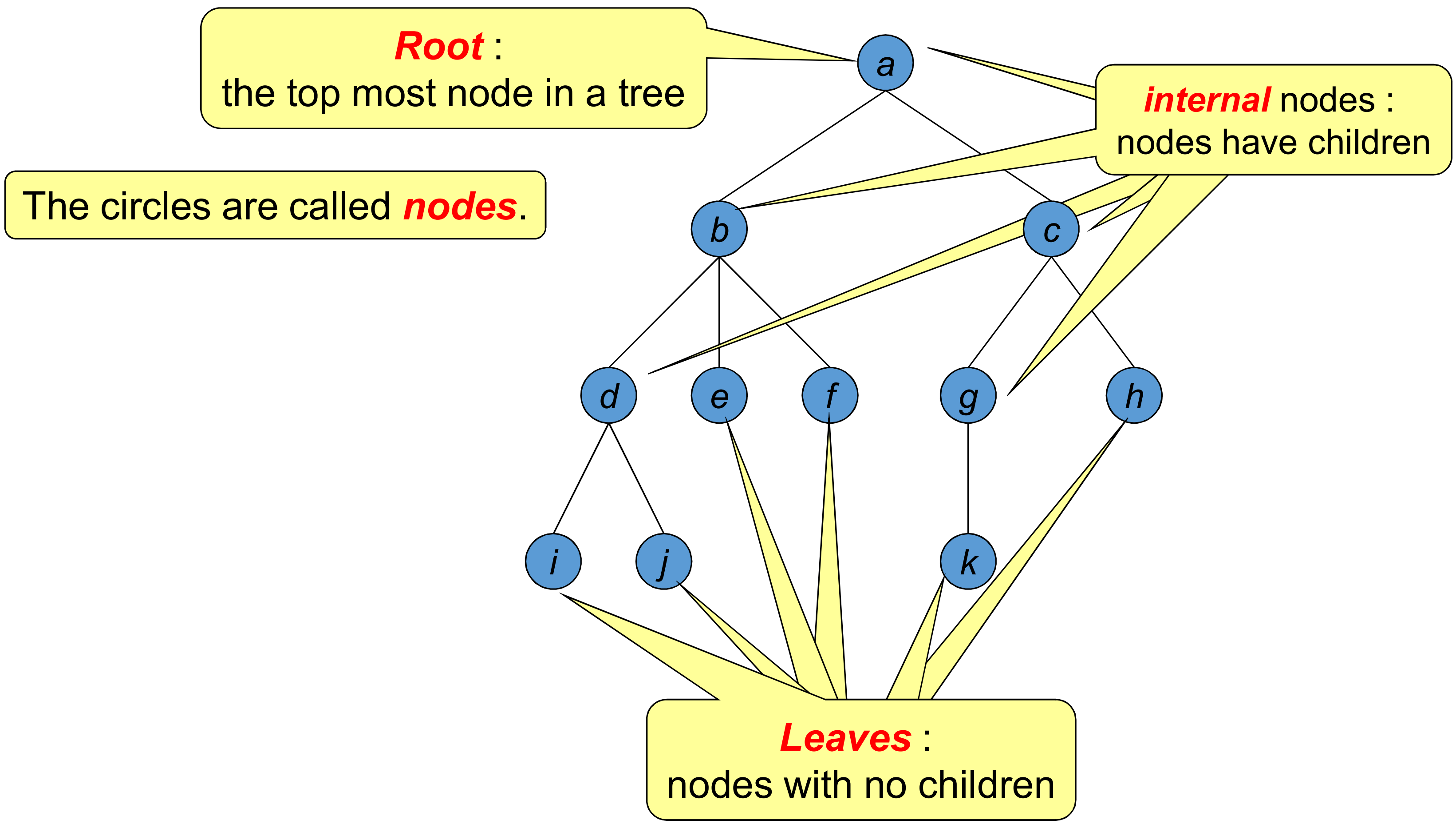
Tree Terminologies



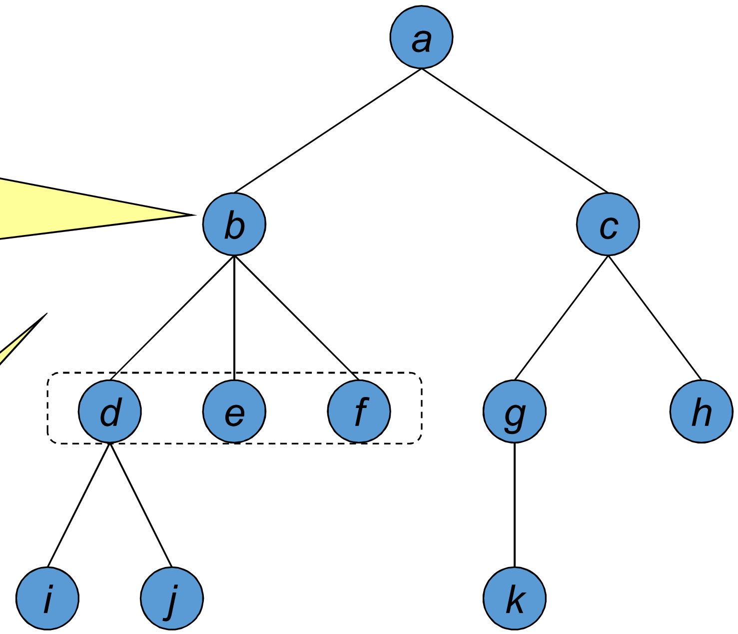
Definition

- A tree is a collection of nodes.
- The collection
  - can be empty;
  - or consists of
    - a distinguished node, called the root,
    - and zero or more non-empty (sub) trees, each of whose roots are connected by a directed edge from the root.

Tree Terminologies



Tree Terminologies



Node *b* is the ***parent*** of nodes *d*, *e*, and *f*.

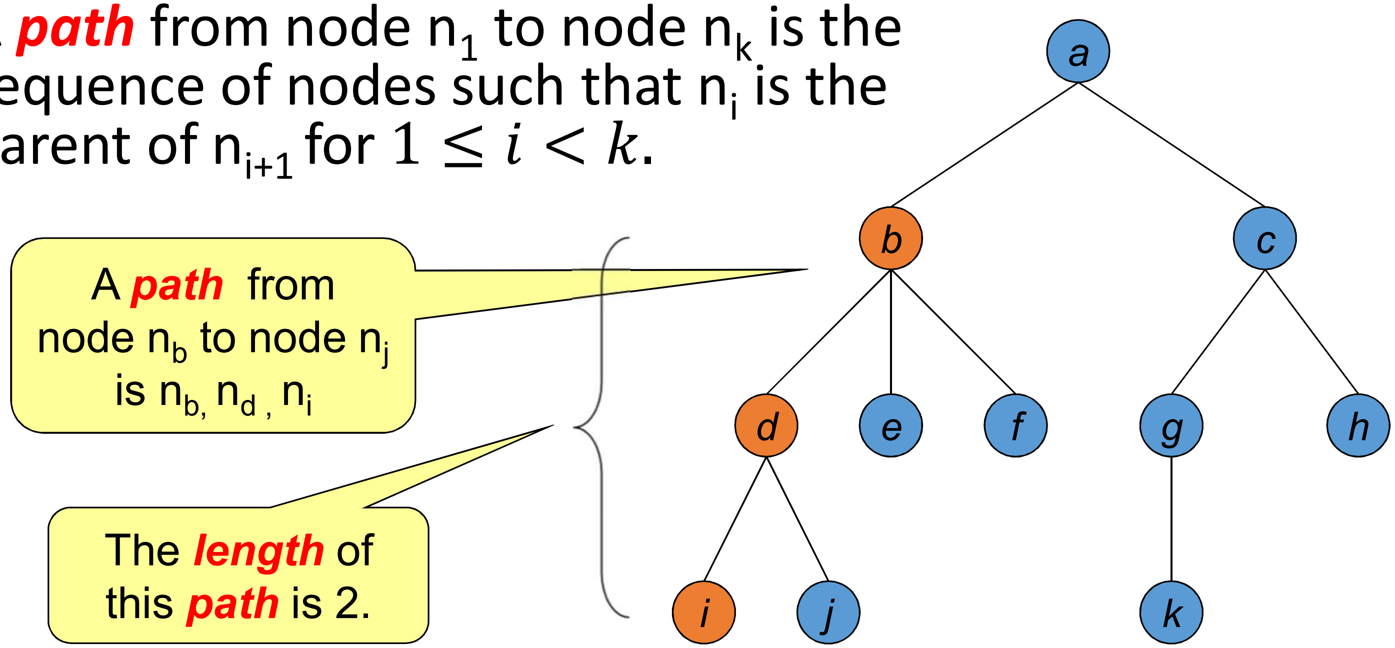
Nodes *d*, *e*, and *f* are the ***children*** of node *b*.

Nodes *d*, *e*, and *f* are ***siblings***. They have the same parent.

ancestor : parent and parent's ancestors
descendant : children and children's descendants

Tree Terminologies

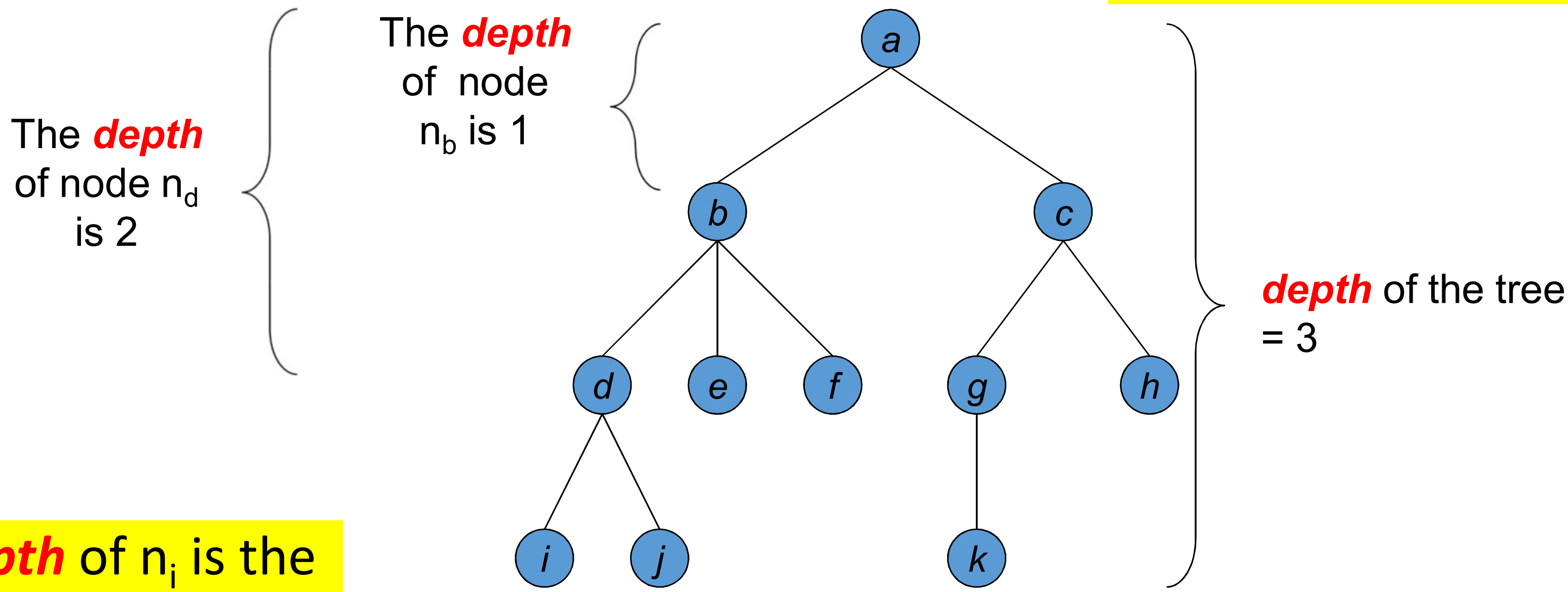
- A ***path*** from node n_1 to node n_k is the sequence of nodes such that n_i is the parent of n_{i+1} for $1 \leq i < k$.



- The ***length*** of this path is the number of edges on the path, namely k-1
- In a tree, there is exactly one path from the root to each node.

Tree Terminologies

- The ***depth*** of a tree = the depth of the deepest leaf.

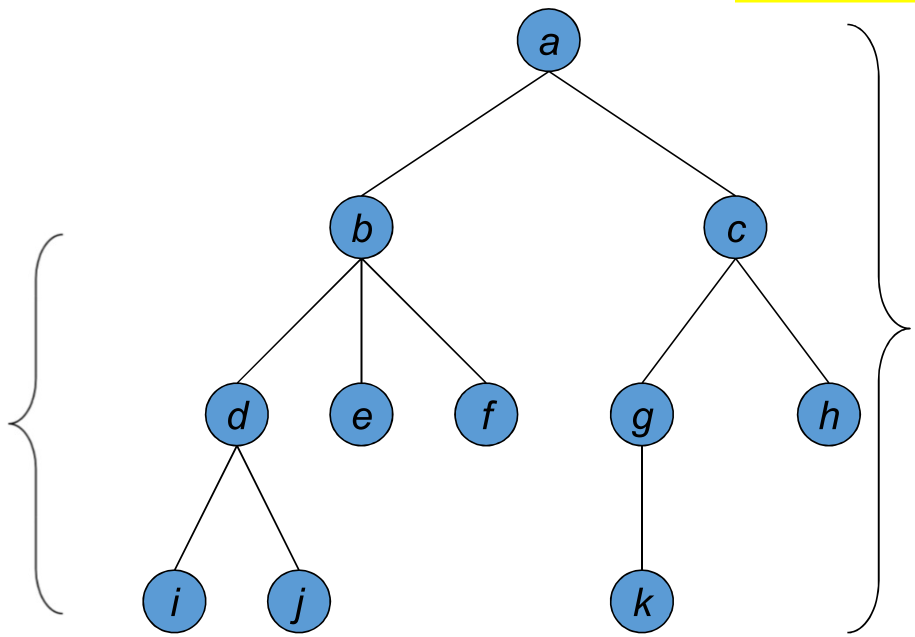


- The ***depth*** of n_i is the length of the unique path from the root to n_i.

Tree Terminologies

- The ***height*** of a tree = the height of the root.

- The ***height*** of n_i is the length of the longest path from n_i to a leaf.

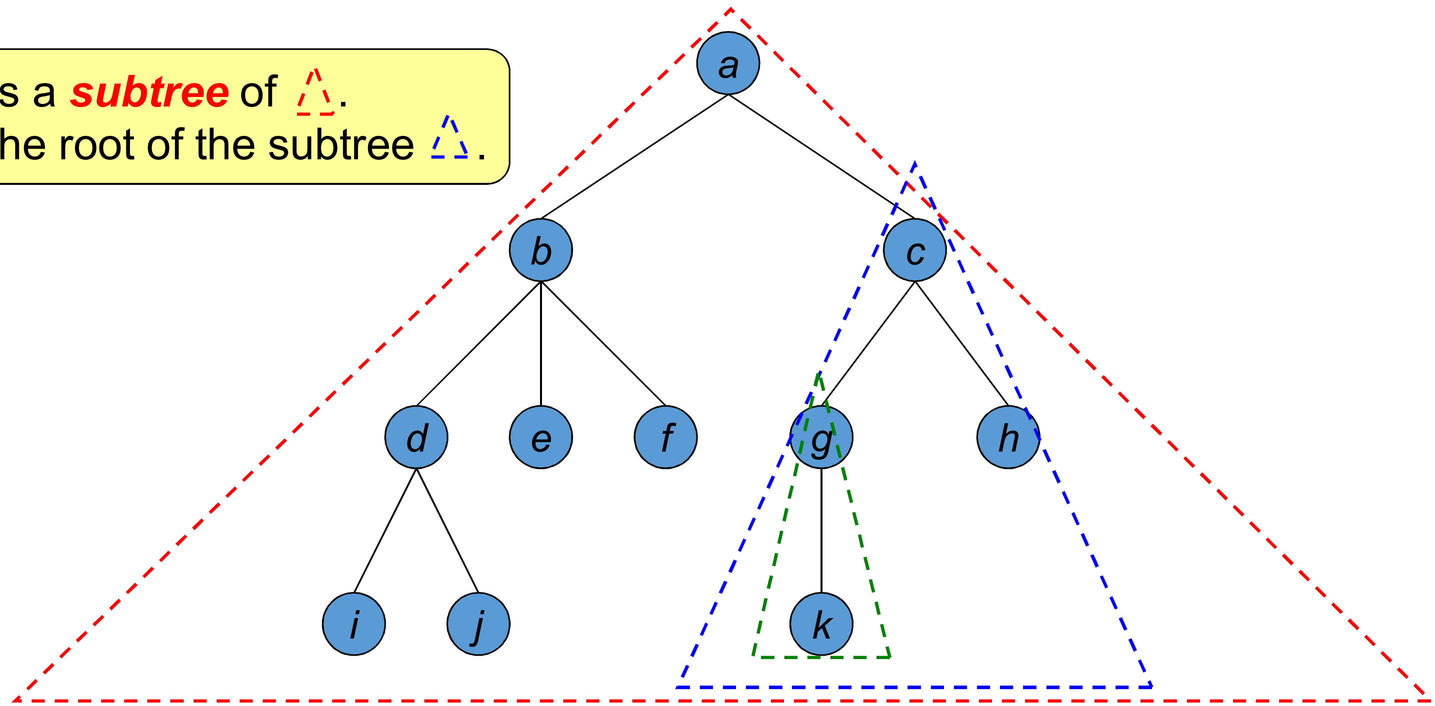


height of the tree = ***depth*** of the tree = 3

A ***height*** of node n_b is 2

Tree Terminologies

is a ***subtree*** of .
Node *c* is the root of the subtree .

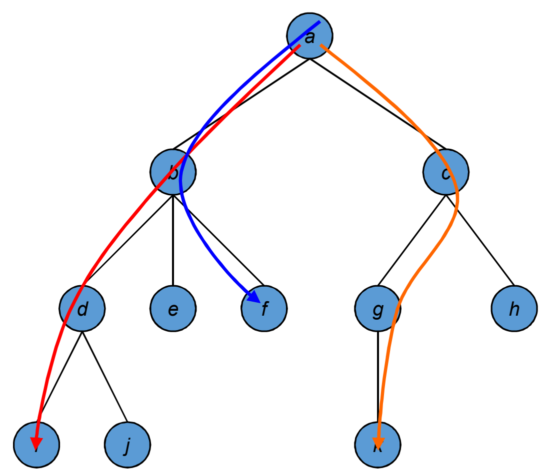


is a ***subtree*** of .
Node *g* is the root of the subtree .

Tree Properties

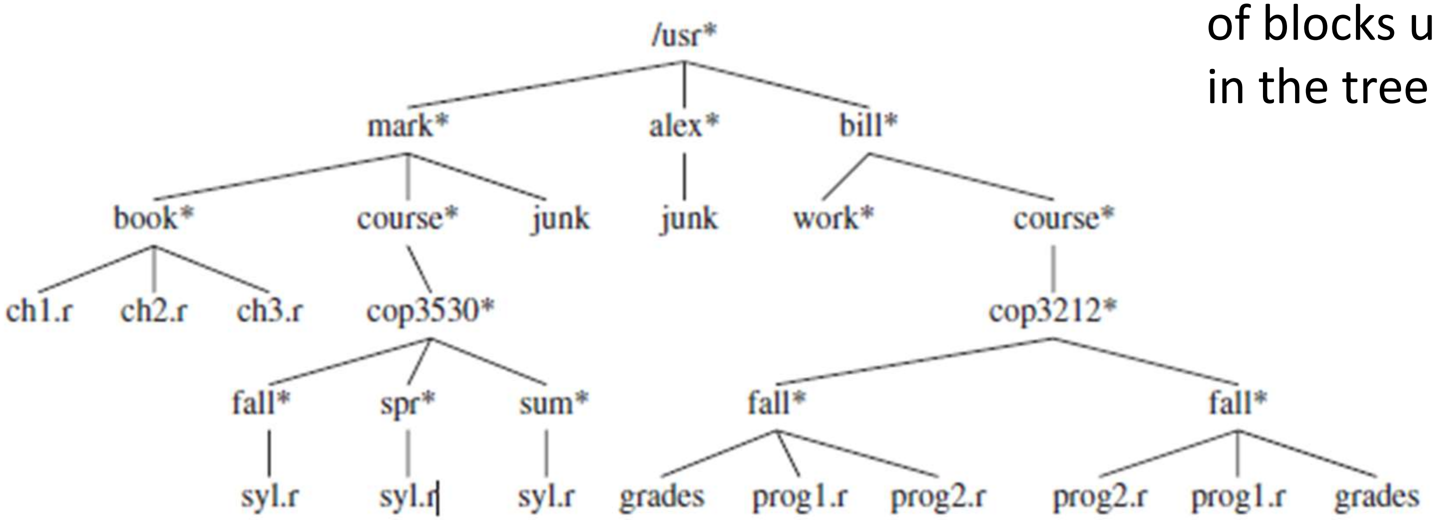
- As long as a tree contains some nodes, there must be a *root* that forms the top of a hierarchy.

- Every other node is connected to the root by a *unique* line of descendants.



Applications

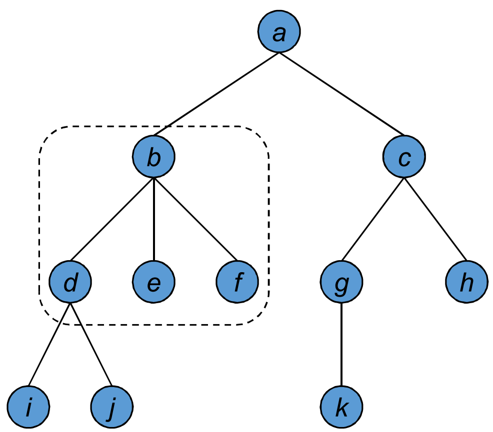
- Directory structure in UNIX etc.



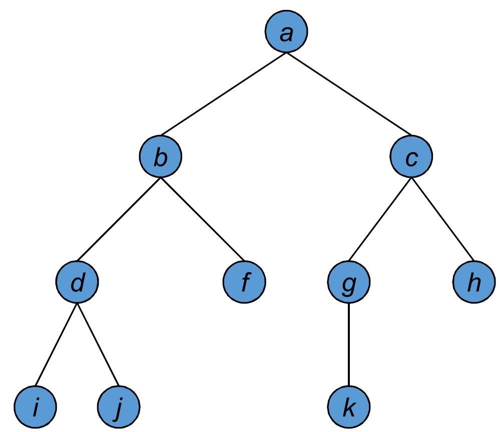
- Listing the names of all files in the directory -- How ?
- Calculating the total number of blocks used by all the files in the tree

Binary Trees

- A ***binary*** tree is a tree in which every node has at most 2 children.



Not a binary tree

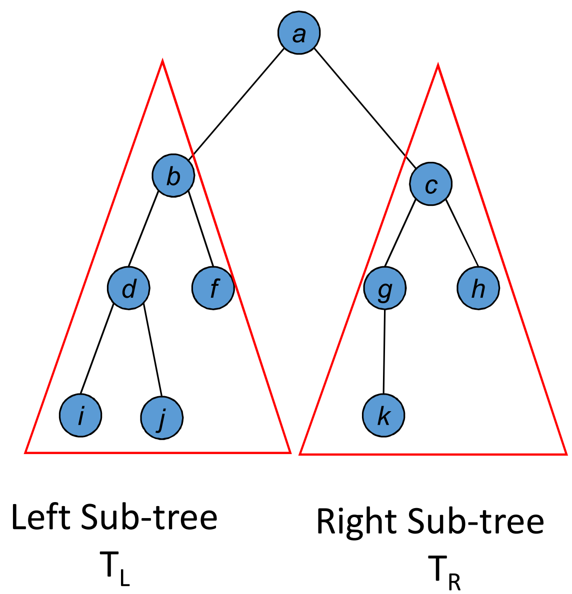


A binary tree

Binary Trees

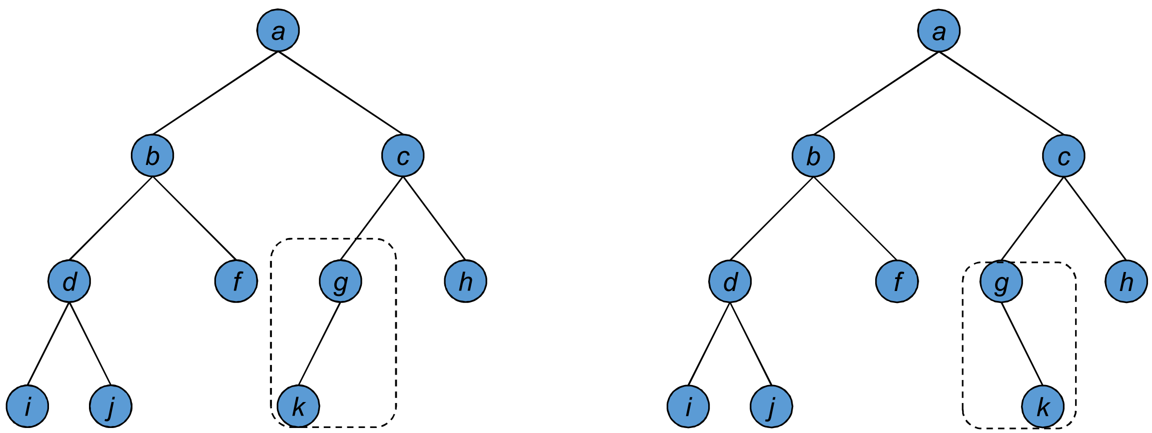
- A ***binary*** tree is a tree in which every node has at most 2 children.

A binary tree



Binary Trees

- In a binary tree, every node except the root is designated as either a ***left child*** or a ***right child*** of its parent.



These two binary trees are *different*.

Application of Binary Trees : expression tree

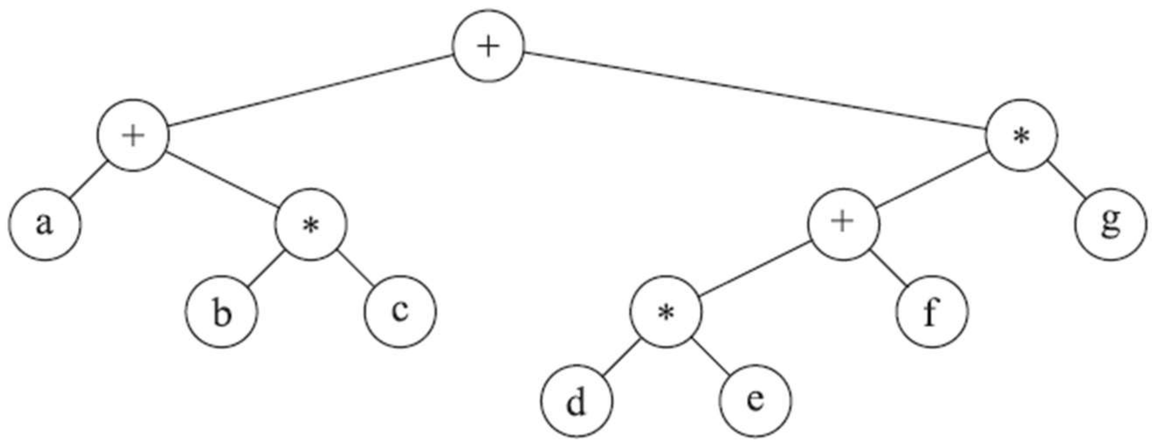


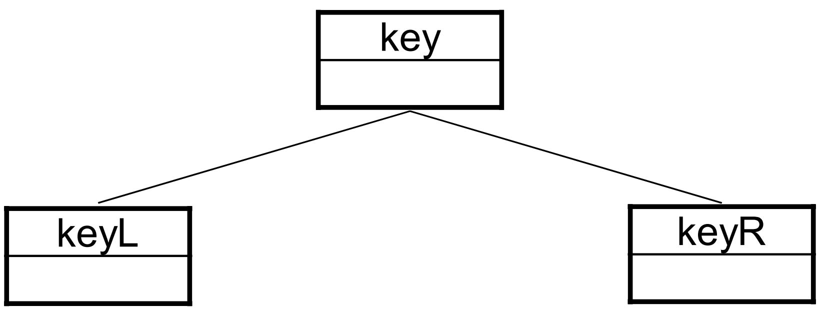
Figure 4.14 Expression tree for (a + b * c) + ((d * e + f) * g)

Reverse Polish notation or Postfix :
a b c * + d e * f + g * +

Traversal : conversion between the expression tree and the notation

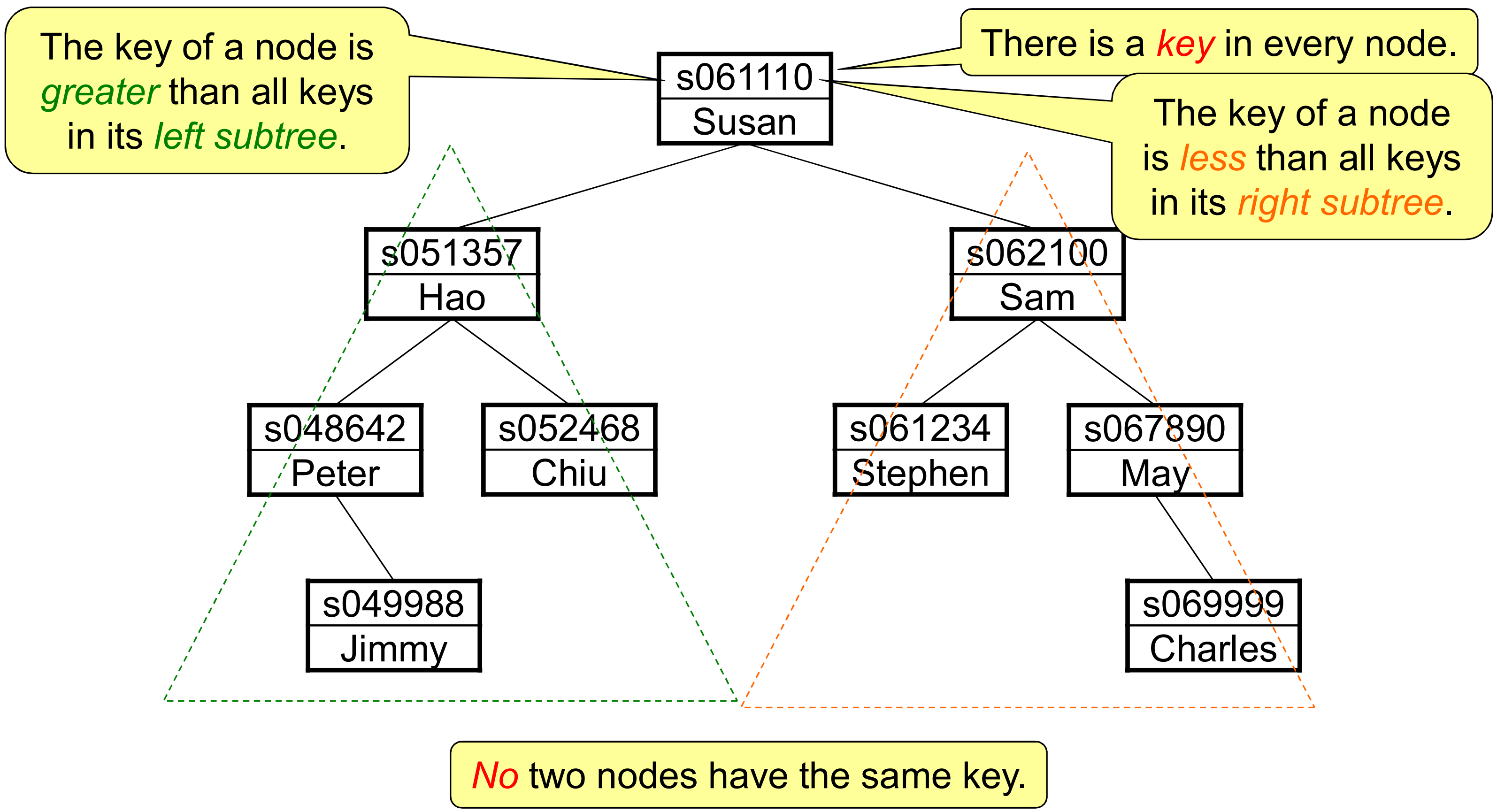
Binary Search Trees

- A ***binary search tree*** (BST) is a binary tree with the following properties.
  - Every node contains a ***key*** that defines the order of the nodes.
  - Keys are *unique* in the tree.
  - At every node in the tree, the key of the node must be
    - *greater* than all the keys in its *left subtree*;
    - *less* than all the keys in its *right subtree*.

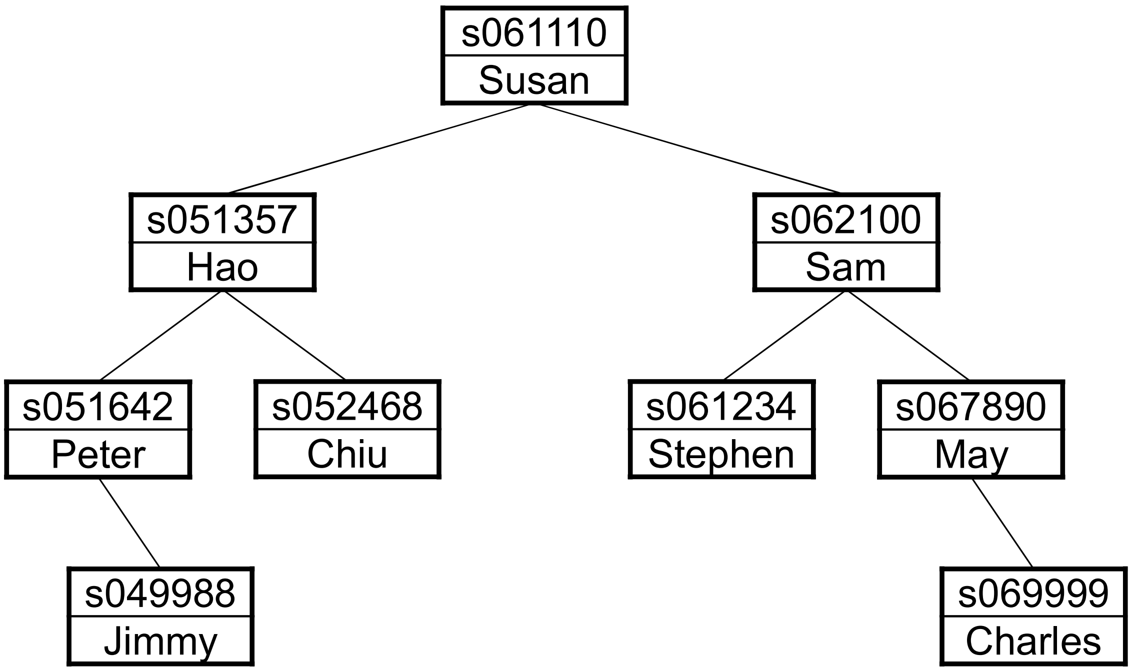


keyL < key < keyR

Binary Search Trees: Example

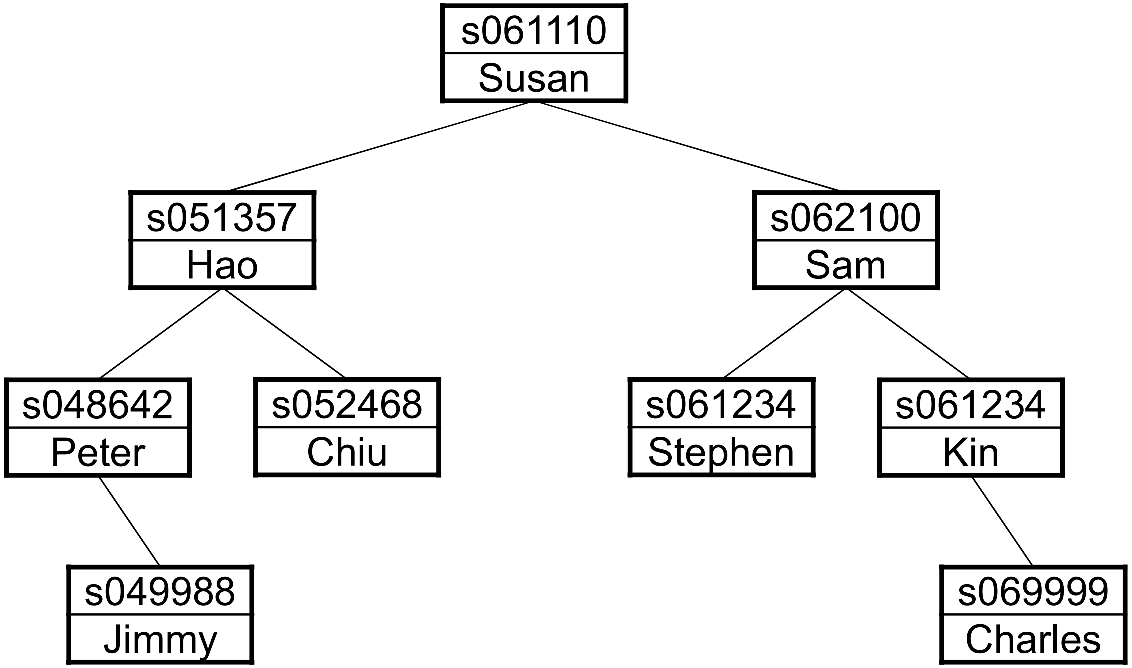


Exercise: Is This a BST?



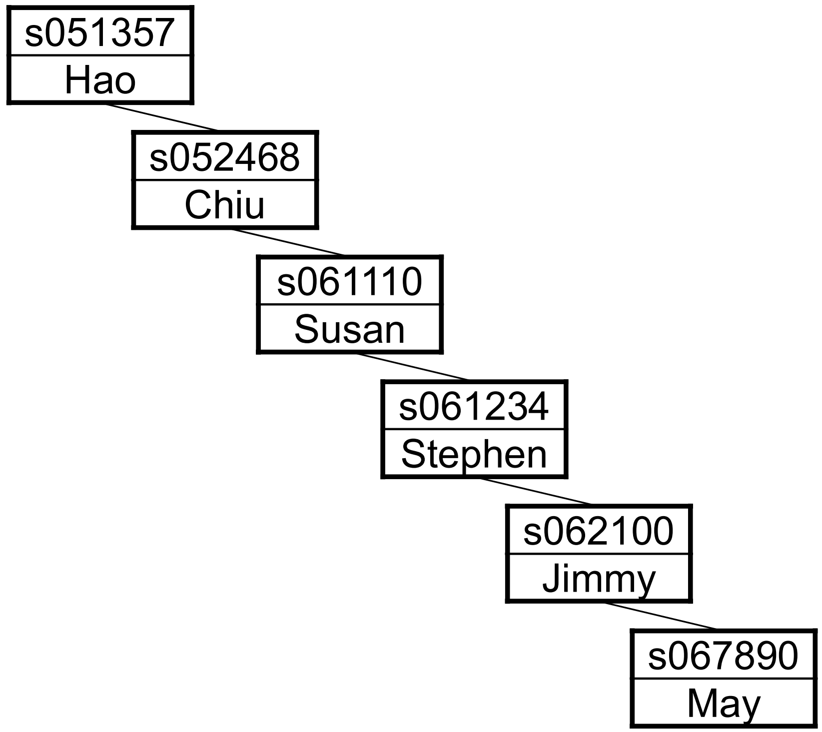
- Every node contains a *unique* ***key***.
- The key of a node must be *greater* than all those in its *left subtree*.
- The key of a node must be *less* than all those in its *right subtree*.

Exercise: Is This a BST?



- Every node contains a *unique* ***key***.
- The key of a node must be *greater* than all those in its *left subtree*.
- The key of a node must be *less* than all those in its *right subtree*.

Exercise: Is This a BST?



- Every node contains a *unique* ***key***.
- The key of a node must be *greater* than all those in its *left subtree*.
- The key of a node must be *less* than all those in its *right subtree*.