1. **Binary trees**. Each node has 2 children which can be empty. It has also a parent which also can be empty.

A node without any children (both empty) is leaf. Other nodes are internal. The node without any parent is root.

A Pascal type representation:

```pascal
type node = record
  k, l, r : node;
  key : integer;
end

tree = node
```

2. **More terminology.** If $y$ is a descendant of $u$ if it is a child of $u$ or a descendant of a child of $u$.

If $y$ is descendant of $u$, then $u$ is an ancestor of $y$.

- Depth of $u$ is number of edges from root to $u$.
- Height of $u$ is max. number of edges from $u$ to a leaf.
- Subtree of $u$ consists of $u$ and its descendants.
3. Traversals. Three traversals are inorder, preorder and postorder.

Procedure Inorder (M)
    if M \neq nil then
        Inorder (M.l);
        print (M.key);
        Inorder (M.r);
    endif

Procedure Preorder (M)
    if M \neq nil then
        print (M.key);
        preorder (M.l);
        preorder (M.r);
    endif

Postorder: first recurse left, then right, then print
property: \( m \) is ancestor of \( y \) iff
\[
\text{pre}(m) < \text{pre}(y) \quad \text{and} \quad \text{post}(m) > \text{post}(y)
\]

A binary tree is a binary search tree if the keys printed inorder is sorted.
4. Searching

```plaintext
function Search(x; μ)
    if μ = nil or x = μ.key then return μ
    else if x < μ.key then return Search(x, μ.l)
    else return Search(x, μ.r)
    endif
    endif
```

Write Successor and Min functions.
5. Insertion and Deletion

procedure Insert (P, Y)
    \( x := \text{nil} \); \( M := P \);
    while \( M \neq \text{nil} \) do
        \( x := Y \);
        if \( Y.\text{key} < M.\text{key} \) then
            \( M := M.\text{l} \);
        else
            \( M := M.\text{r} \);
        endif
    endwhile

    \( Y.\text{p} := x \);
    if \( x = \text{nil} \) then \( P := Y \)
    else if \( Y.\text{key} < x.\text{key} \) then
        \( x.\text{l} := Y \)
        \( x.\text{r} := X \);
    elif \( Y.\text{key} > x.\text{key} \) then
        \( x.\text{r} := Y \);
        \( x.\text{l} := X \);
    endif

endif

Deletion is slightly complicated:

Case 1: node has no children
Case 2: node has one child
Case 3: node has two children
  a. replace key by Successor key
  b. delete Successor