Binary Tree Representation

What are the rooted tree representations?

- Single array representation
  - Complete Binary Tree
- Linked List representation

Representation of Rooted Trees

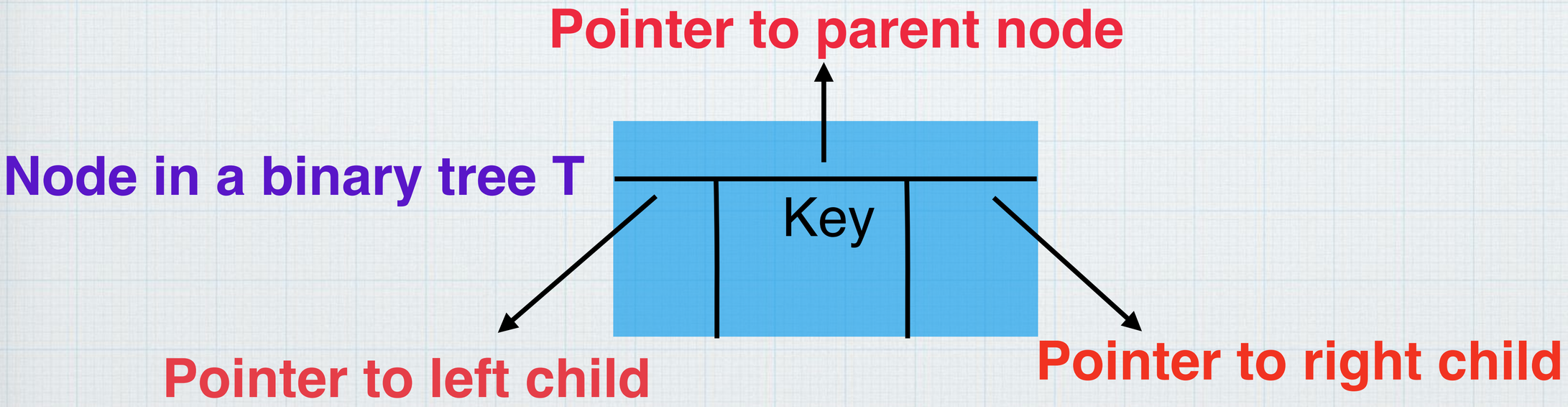
- Representing rooted trees by **linked data structures**
  - Binary trees
  - Rooted trees, in which nodes having **arbitrary number of children**

Node in a Rooted Tree

- Represent **each node** of a tree **by an object**
  - Each **node** contains a **key attribute**
  - **Pointers** to other nodes (vary according to the type of trees)

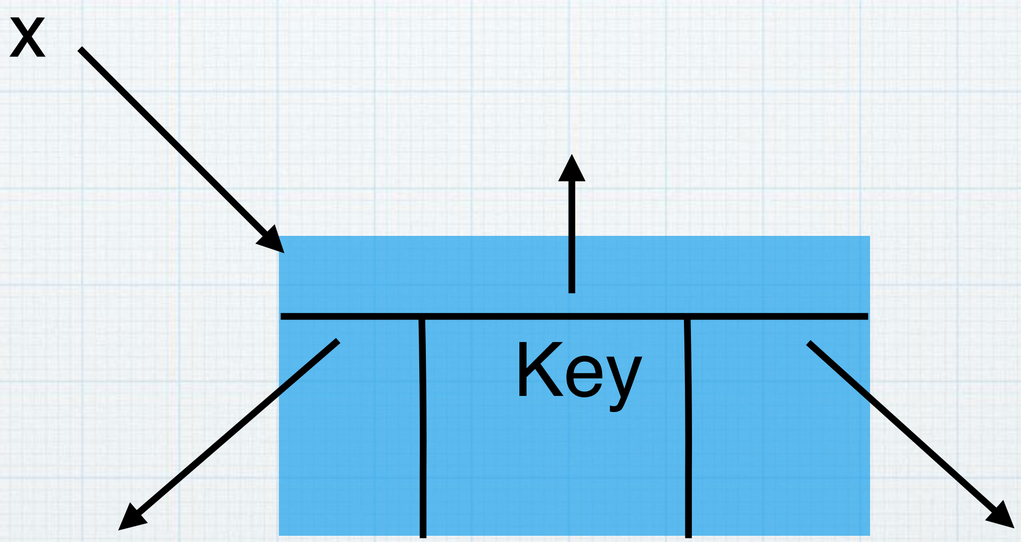
Binary Trees

- Represent **each node** of a tree **by an object**
  - Each **node** contains a **key attribute**
  - **Pointers** to **Parent node, left child and right child**



Special nodes in the Binary tree

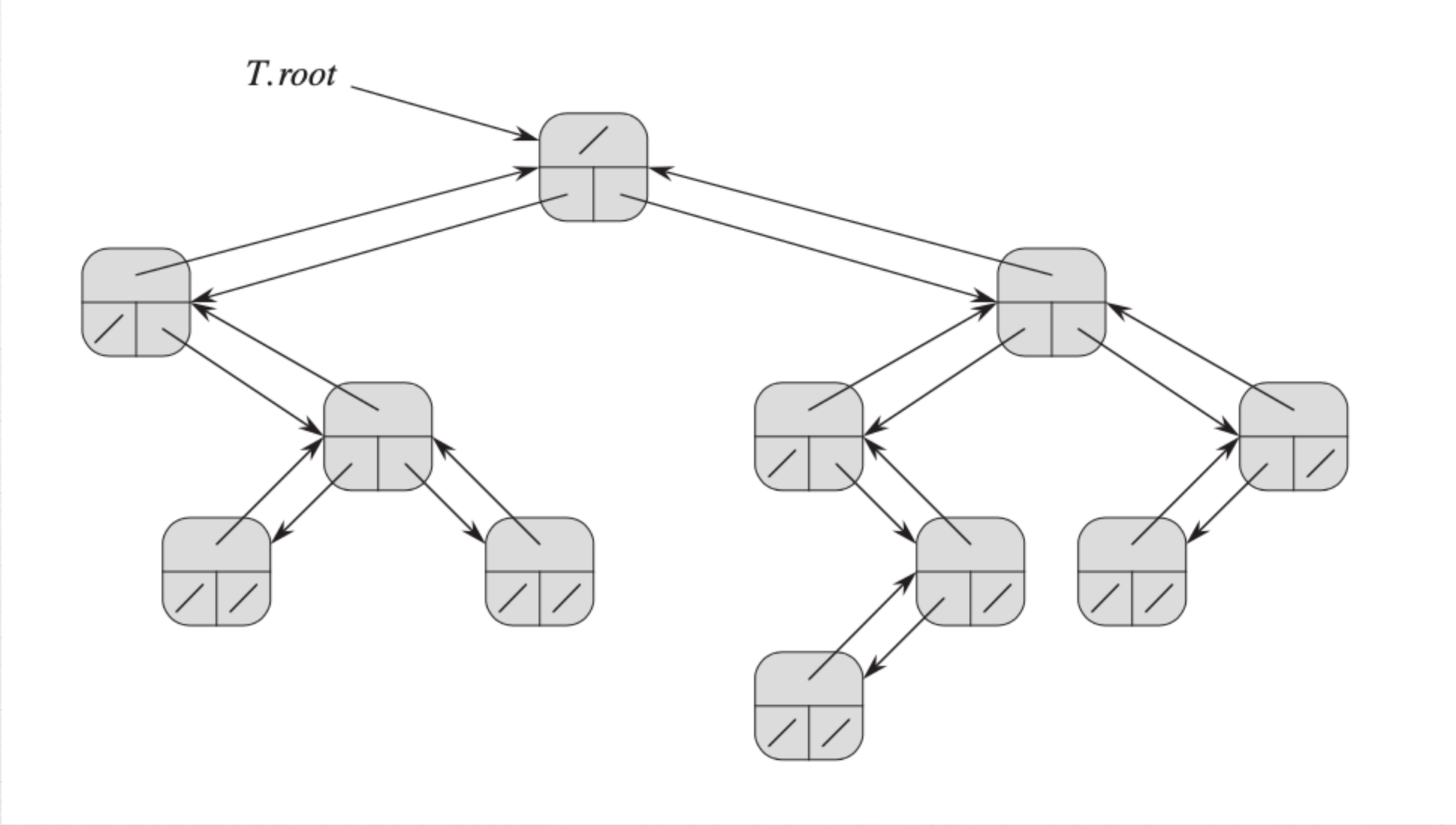
- Node x
- x.p = NIL ——> ?
- x.left = NIL
- x.right = NIL
- If both x.left and x.right are NIL, then ?



Root of the Binary Tree

- The **root of the entire tree T** is pointed to by the attribute **T.** ***root*****.**
- If **T.** ***root*** **= NIL**, then the **tree is empty.**

Representation of a Binary Tree

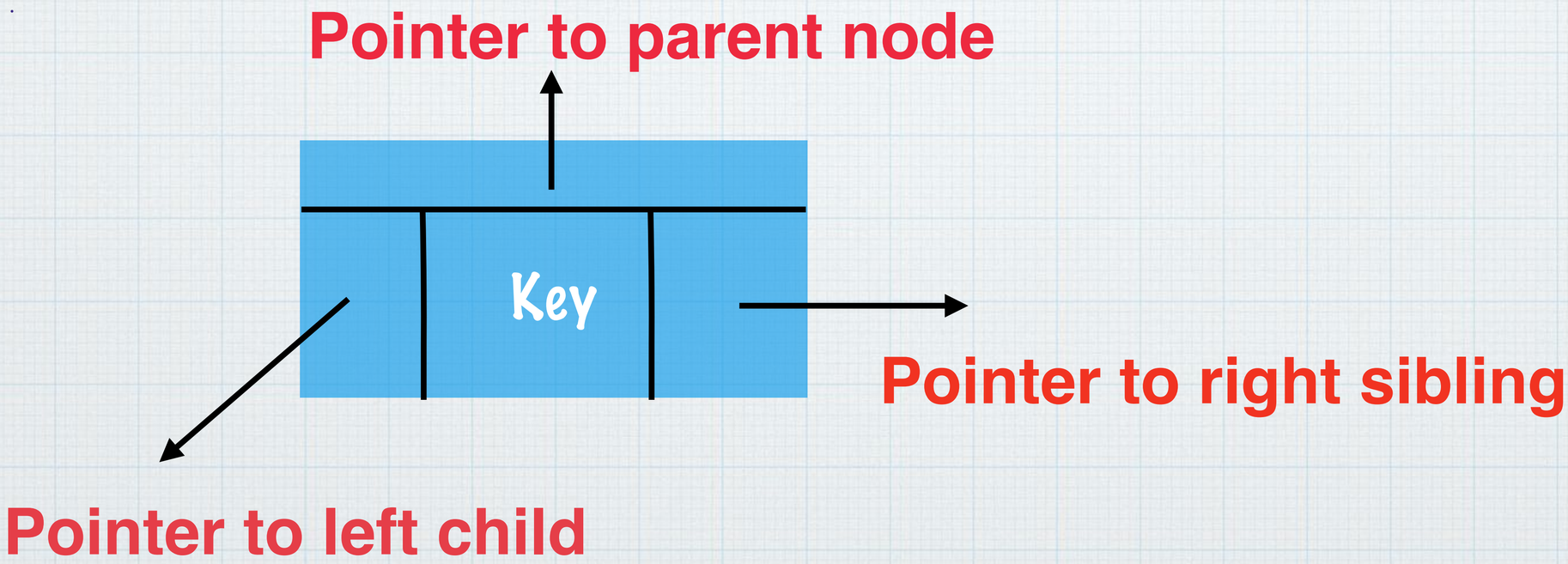


Rooted trees with unbounded branching

- Extend the scheme of **representation of a binary tree** to **any class** of trees
- Trees in which the **number of children of each node** is **at most constant k**
- Replace the *left* and *right* attributes by $child_1, child_2, \ldots, child_k$
- Whether this scheme works, if the number of children of a node is unbounded ?
- Space requirement ?
- Even if the number of children k is bounded by a large constant but most nodes have a small number of children

Left-child, right-sibling representation

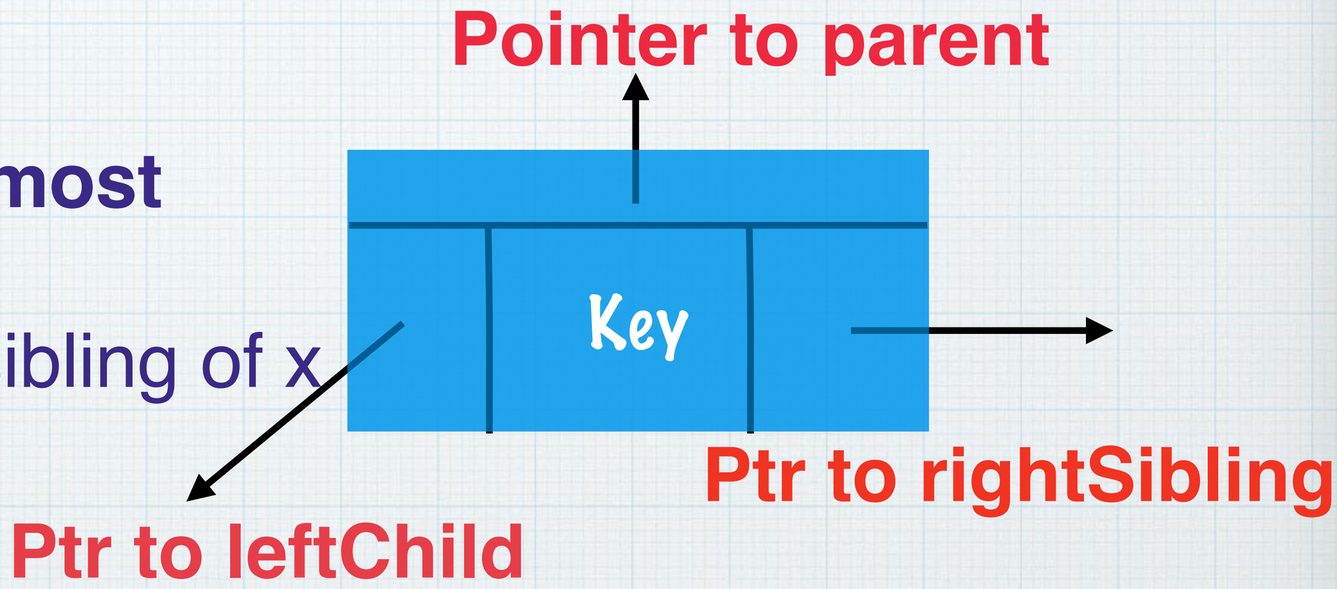
- Scheme to represent trees with arbitrary numbers of children
  - **The left-child, right-sibling representation**
- Each node contains a parent pointer p, and **T.*root*** points to the root of tree T.



Left Child, Right Sibling Representation

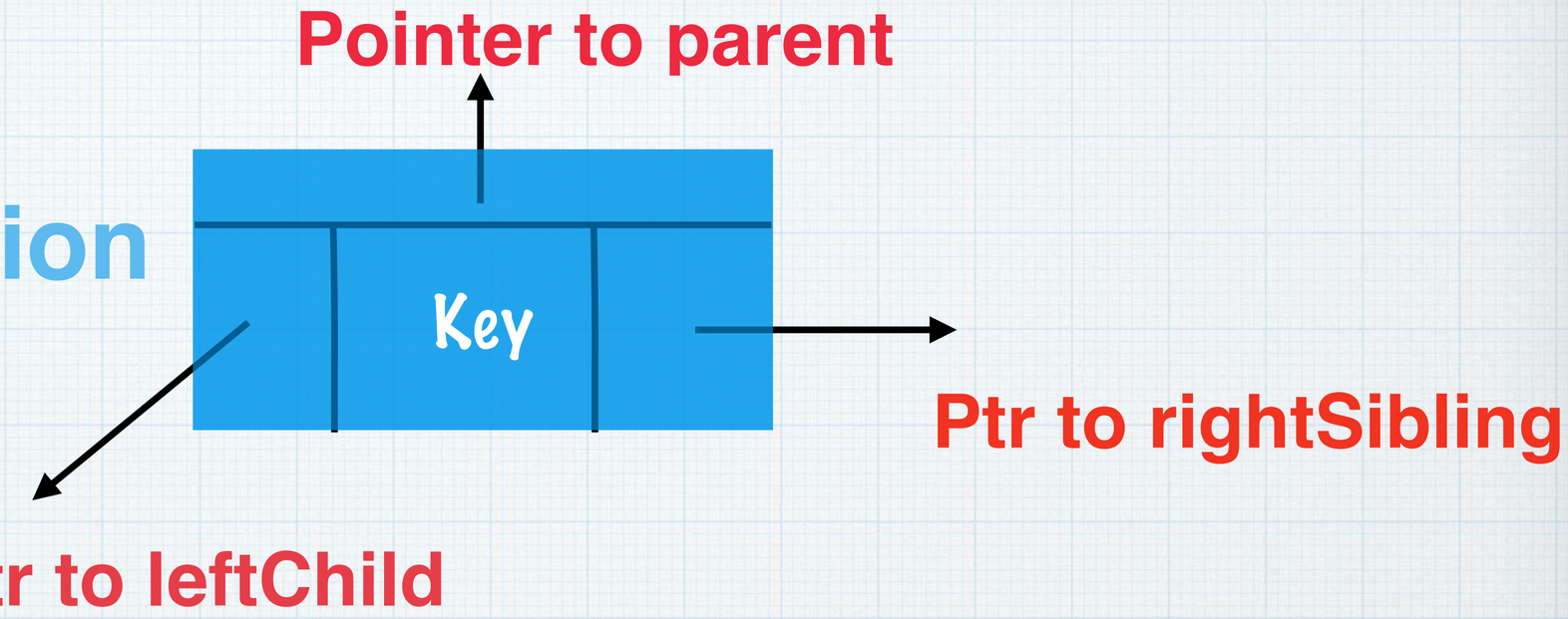
- Instead of having a pointer to each of its children, each **node x** has only **two pointers**:

- 1. **x.*leftChild*** points to the **left most child of node x**
  2. **x.*rightSibling*** points to the sibling of x immediately to its right.



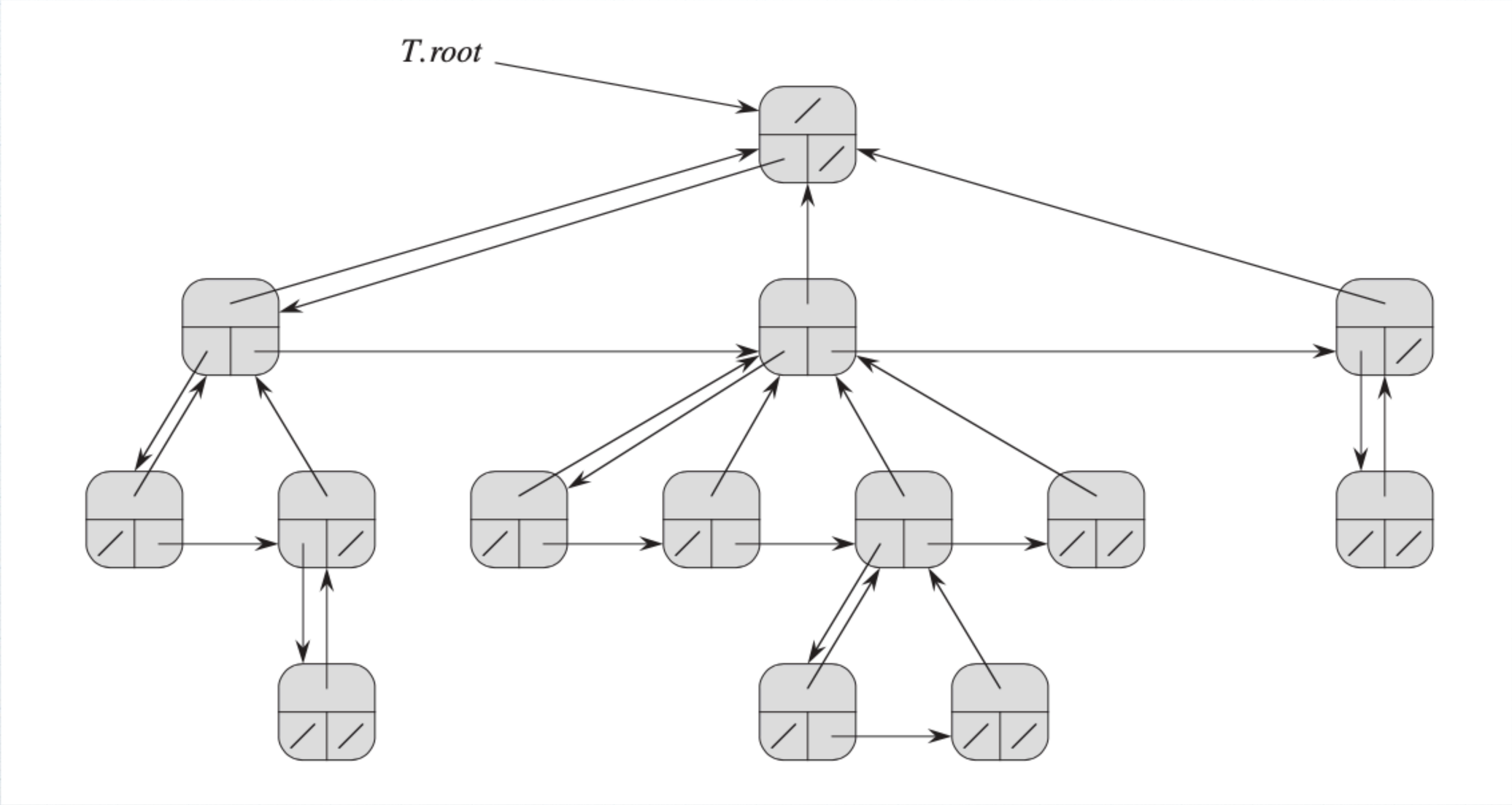
- If **node x** has no children, **x.*leftChild* = NIL**
- **I**f node x is the rightmost child of its parent, then **x.*rightSibling* = NIL**.

Left Child, Right Sibling Representation



- Node x
- x.p = NIL and x.rightSibling = NIL ——> ?
- If both x.leftChild and x.rightSibling are NIL ——> ?

Left-child and Right-Sibling Representation



Implementation Details

```
struct node
{
        elemType Key;
        struct node *p;
        struct node *left;
        struct node *right;
};
```

```
struct binaryTree
{
 struct node *root;
};
```

Reference

* CLRS Book