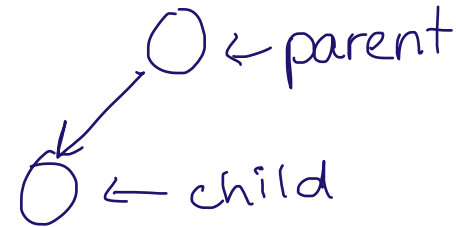


terminology

- ▶ set of nodes (possibly with values or labels), with directed edges between some pairs of nodes
- ▶ One node is distinguished as root
- ▶ Each non-root node has exactly one parent. (we call this node the parent's child)
- ▶ A path is a sequence of nodes n_1, n_2, \dots, n_k , where there is an edge from n_i to n_{i+1} . The length of a path is the number of edges in it
- ▶ There is a unique path from the root to each node. In the case of the root itself this is just n_1 , if the root is node n_1 .
- ▶ There are no cycles — no paths that form loops.



more terminology

- ▶ leaf: node with no children

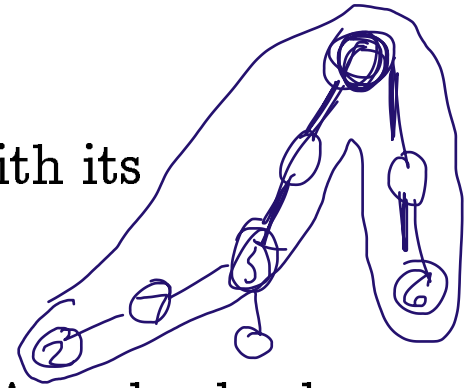
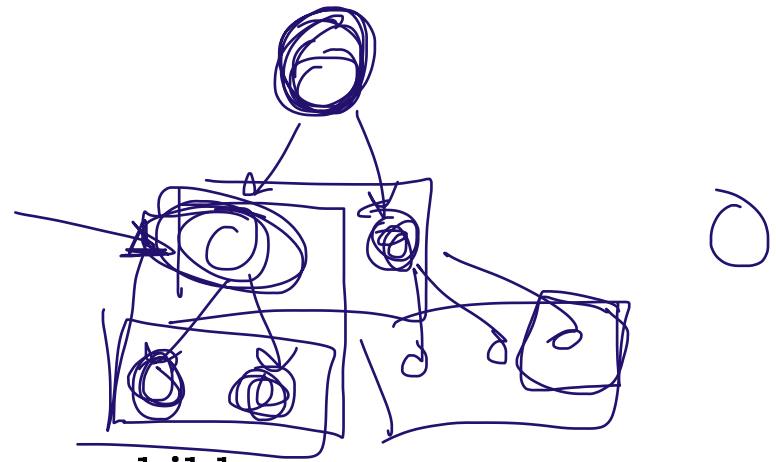
- ▶ internal node: node with one or more children

- ▶ subtree: tree formed by any tree node together with its descendants and the edges leading to them.

- ▶ height: $1 + \text{the maximum path length in a tree}$. A node also has a height, which is $1 + \text{the maximum path length of the tree rooted at that node}$

- ▶ depth: length of a path from root to a node is the node's depth.

- ▶ arity, branching factor: maximum number of children for any node.



+1?

general tree implementation

```
class Tree:
    """
    A bare-bones Tree ADT that identifies the root with the entire tree
    """
    def __init__(self, value=None, children=None):
        """
        Create Tree self with content value and 0 or more children

        @param Tree self: this tree
        @param object value: value contained in this tree
        @param list[Tree] children: possibly-empty list of children
        @rtype: None
        """
        self.value = value
        # copy children if not None
        self.children = children.copy() if children else []
        if len(children) > 0:
            self.children = children.copy()
        else:
            self.children = []
```

equiv

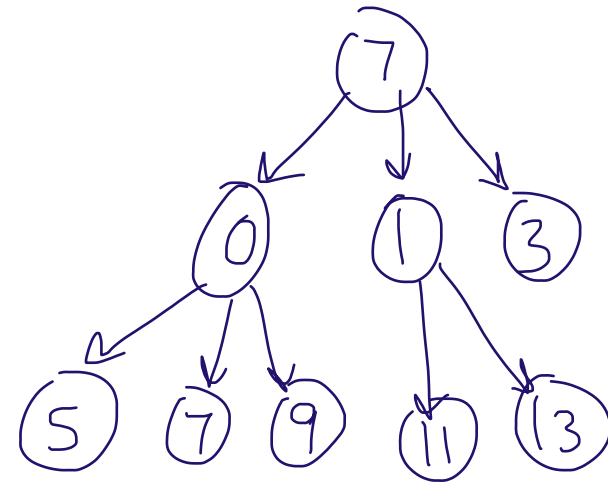
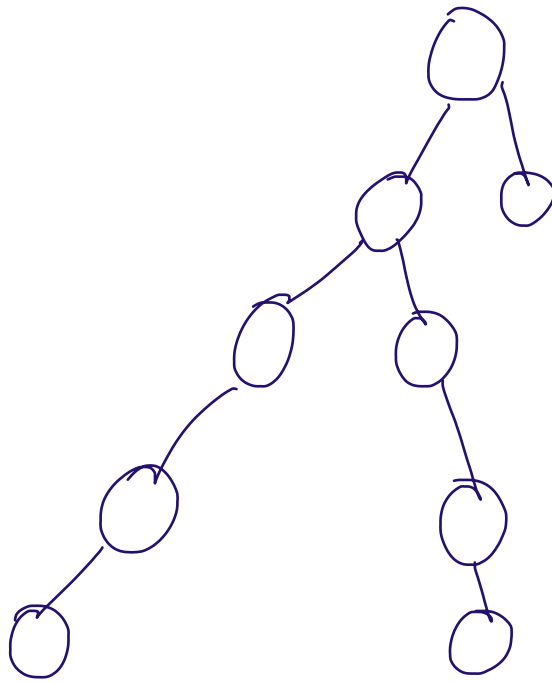
general form of recursion:

if \langle condition to detect a base case \rangle : \rightarrow commonly, this
Tree is a leaf

\langle do something without recursion \rangle

else: # \langle general case \rangle

\langle do something that involves recursive call(s) \rangle



descendants from list (Tree(7), [0, 1, 3], [5, 7, 9], [11, 13], 3)

