CSE 101-02 5-12-22

- mid 2 : Thur May 19
- PA6 : ext 1 day (Tue.)

- PA6 : Question

from-string constructor

Dictionary ADT (also called Map)

State: a set of ordered pairs

(Key, Value)

i.e. a state is

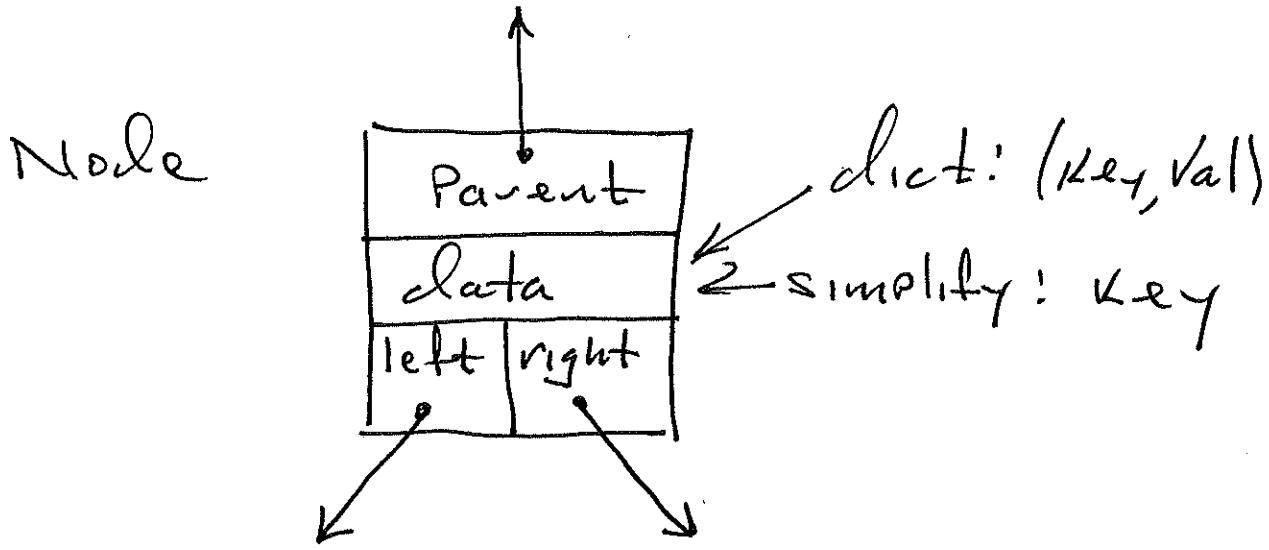
$$\{(K_1, V_1), (K_2, V_2), \ldots, (K_n, V_n)\}$$

Option: Keys are unique

Operations: Main 3

- insert(K, V): inserts a new pair in dictionary. Pre: K does not exist in dict. (lookup == nil)
- lookup(K): returns V s.t. (K, V) is in dict., or nil if no such pair exists.
- delete(K): removes pair (K, V). Pre: lookup(K) ≠ nil.

Binary Search trees (BST)



BST Properties:

let x, y be nodes.

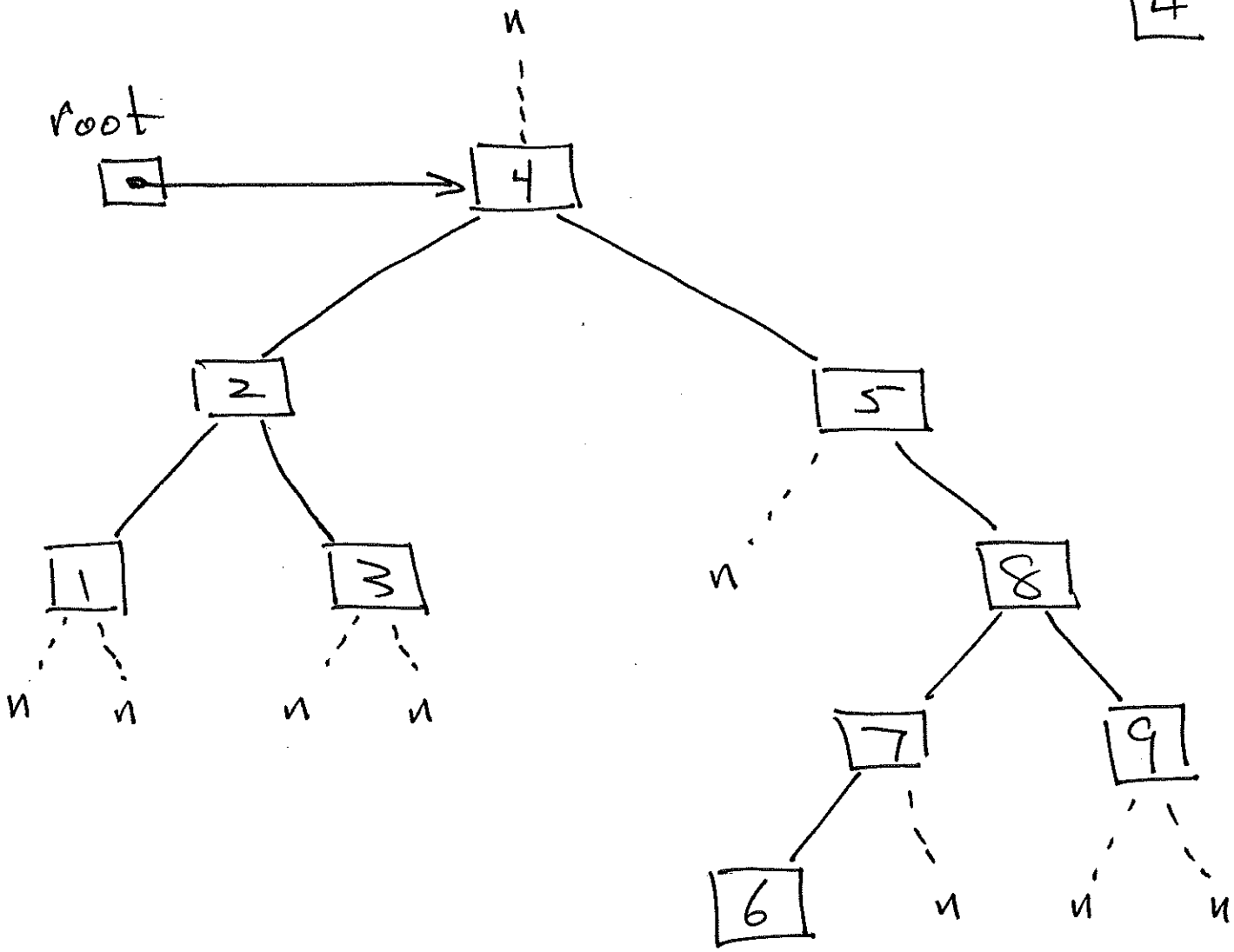
(1) if y is in the left subtree of x

then $key[y] \leq key[x]$

(2) if y is in the right subtree of x

then $key[x] \leq key[y]$.

Ex



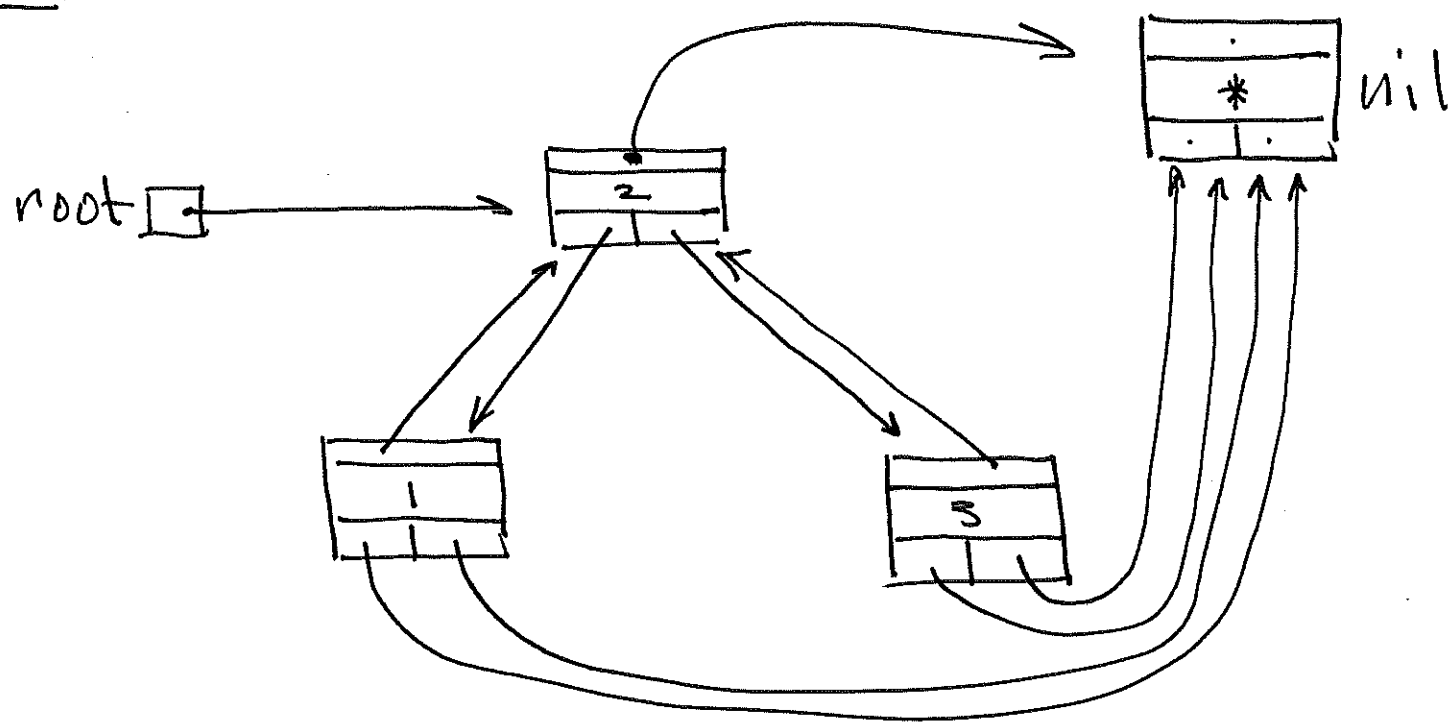
A leaf is a node with no children

An internal node is a non-leaf

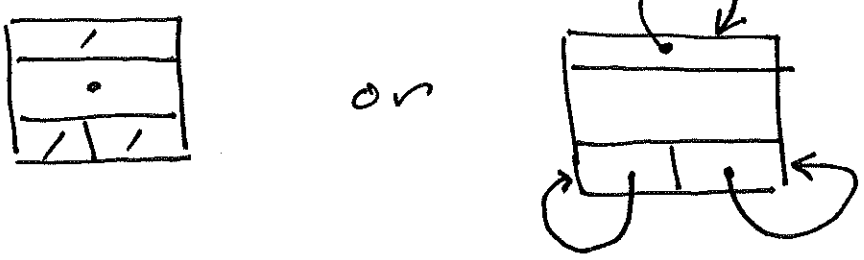
implementation option:

use a dummy (sentinel) node called nil

Ex.



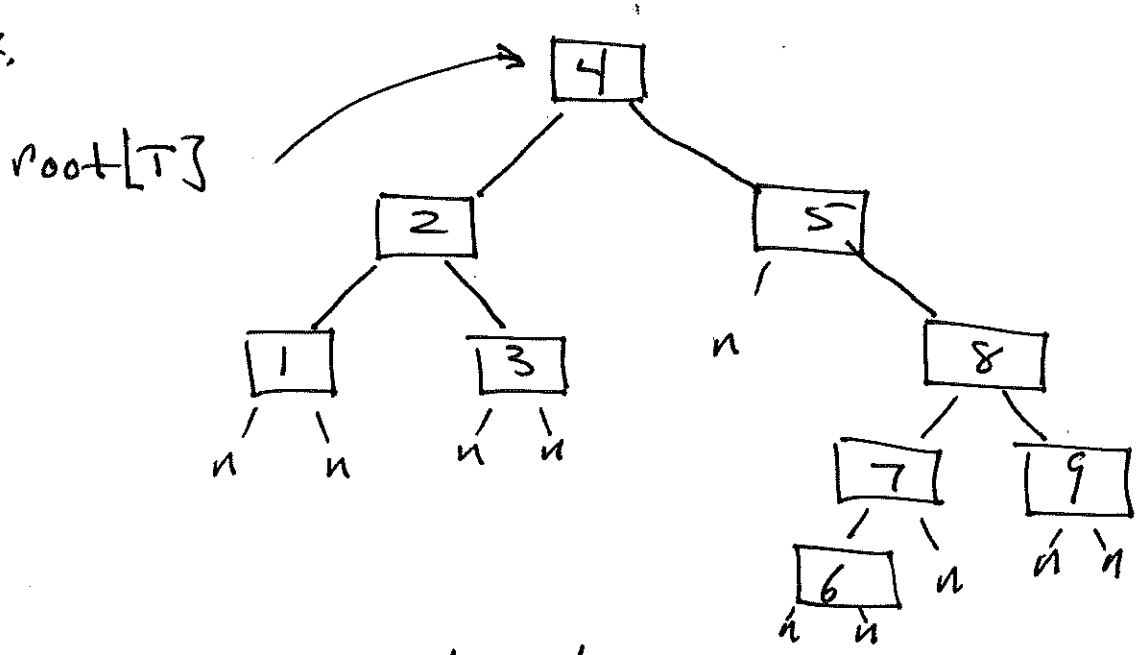
could have



Tree Traversals

- InOrderTreeWalk(x) ← node
- PreOrderTreeWalk(x)
- PostOrderTreeWalk(x)

Ex.



InOrderTreeWalk(root[T])

1 2 3 4 5 6 7 8 9

On same example:

PreOrderTreeWalk(root[T]):

4 2 1 3 5 8 7 6 9

PostOrderTreeWalk(root[T]):

1 3 2 6 7 9 8 5 4

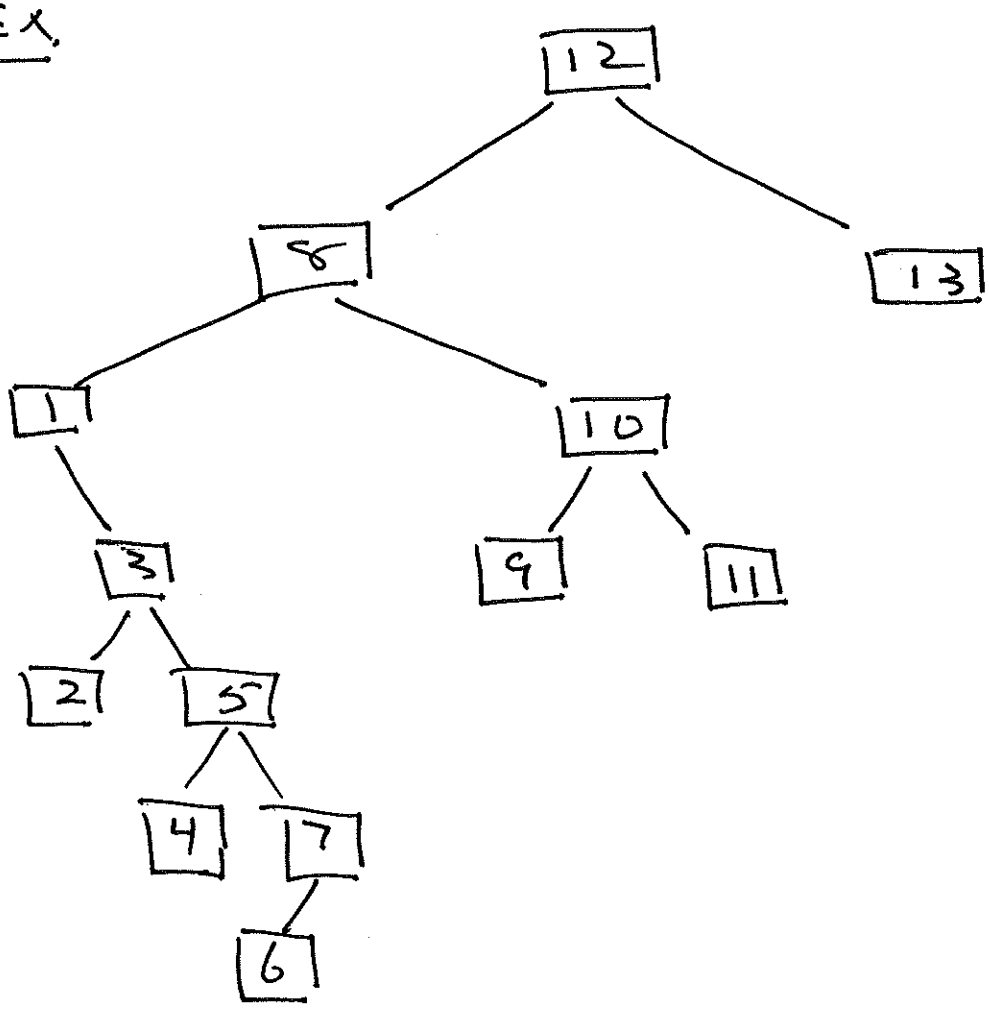
Queries

- TreeSearch(x, k) — x: node
- TreeMinimum(x)
- TreeMaximum(x)

Defn

The successor of a node x is the next node after x to be processed in an inorder tree walk.

Ex.



IOTW: 1 2 3 4 5 6 7 8 9 10 11 12 13

Successor(1) = 2

Successor(8) = 9

Successor(7) = 8

2 cases for successor:

<u>case 1</u> : x has a right child

Successor of x is the leftmost descendant of that child.

<u>case 2</u> : x has no right child

Successor of x is the lowest ancestor of x whose left child is also an ancestor of x

<u>Exercise</u>

- Define Predecessor of x.
- Write algorithm Predecessor(x).

Runtimes:

- Traversals: in, pre, post $\Theta(n)$

where $n = \#$ nodes.

- Queries:

Def: height(T) = dist. from root to a deepest leaf

height(x) = height of subtree rooted at x

note: height$(x) \leq$ height(T)