

CSE 101 11-17-20

11

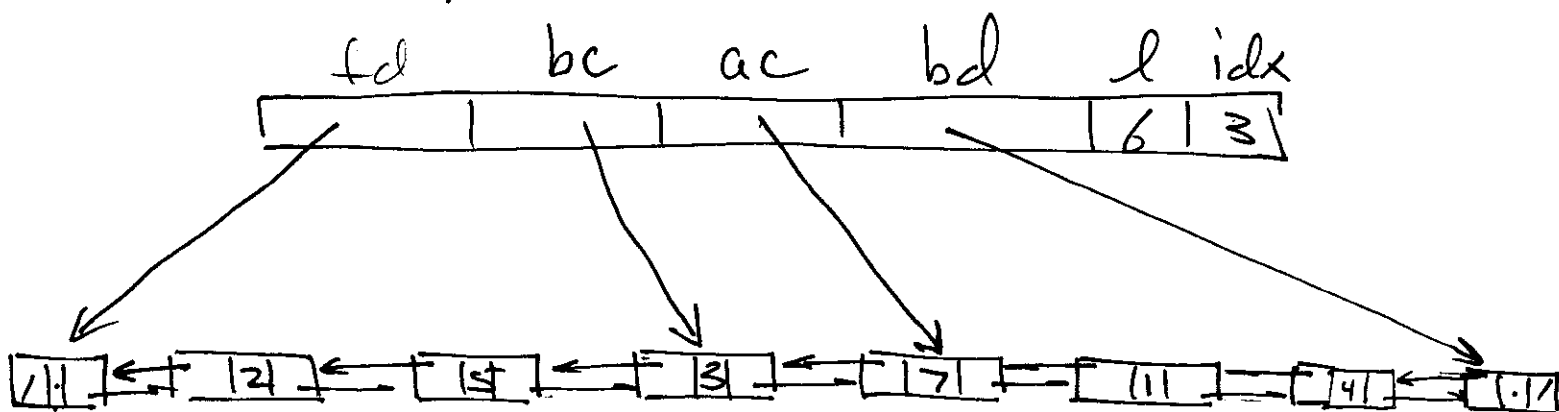
Part: ext. 1 last day

Part 5: how does cursor work?

client view

(⁰ | 2 | ¹ 5 | ² 3 | ³ 7 | ⁴ 1 | ⁵ 4 | ⁶)

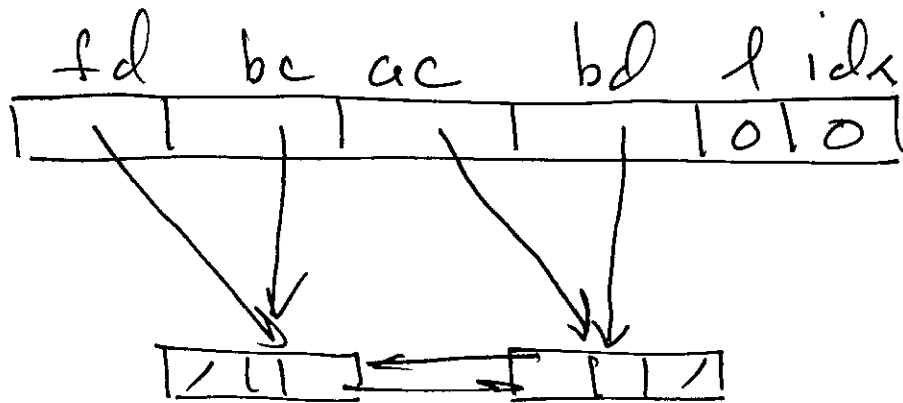
inside view:



empty state:

Client view: $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$

inside view



Continue BSTs ...

tree traversals

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

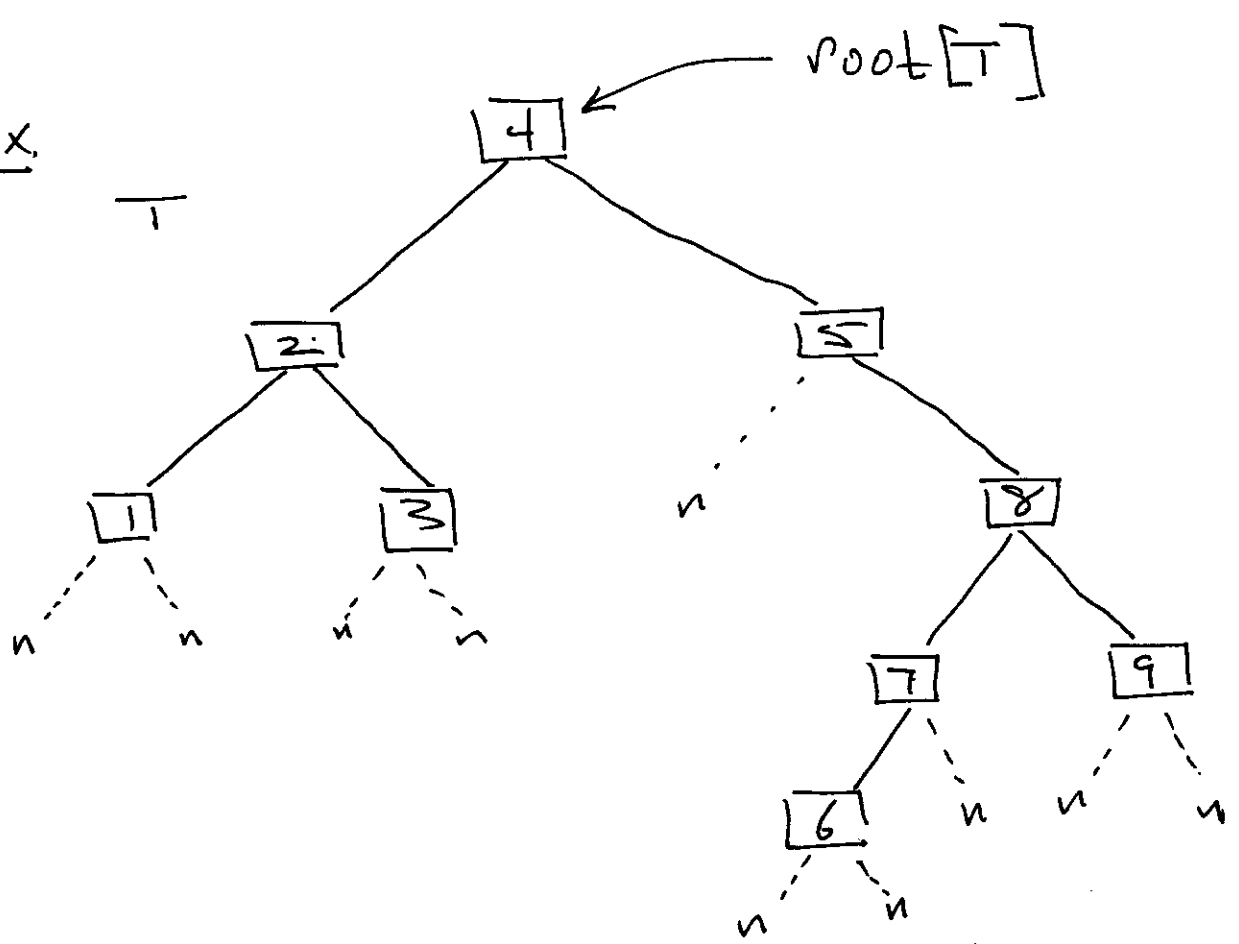
InOrderTreeWalk(x)

1. if $x \neq \text{nil}$
2. InOrderTreeWalk(left[x])
3. Print(key[x]) ←
4. InOrderTreeWalk(right[x])

Book
 $x.\text{left}$
in C
 $x \rightarrow \text{left}$

any processing
of satellite data
at x can occur
here

Ex.



output of InOrderTreeWalk(root[T])

1 2 3 4 5 6 7 8 9

PreOrderTreeWalk(x)

1. if $x \neq \text{nil}$
2. Print (key[x])
3. PreOrderTreeWalk (left[x])
4. PreOrderTreeWalk (right[x])

PostOrderTreeWalk(x)

1. if $x \neq \text{nil}$
2. PostOrderTreeWalk(left[x])
3. PostOrderTreeWalk(right[x])
4. Print(key[x])

Ex. Pre on Prev. example

4 2 1 3 5 8 7 6 9

Ex. Post on Prev. example

1 3 2 6 7 9 8 5 4

Queries

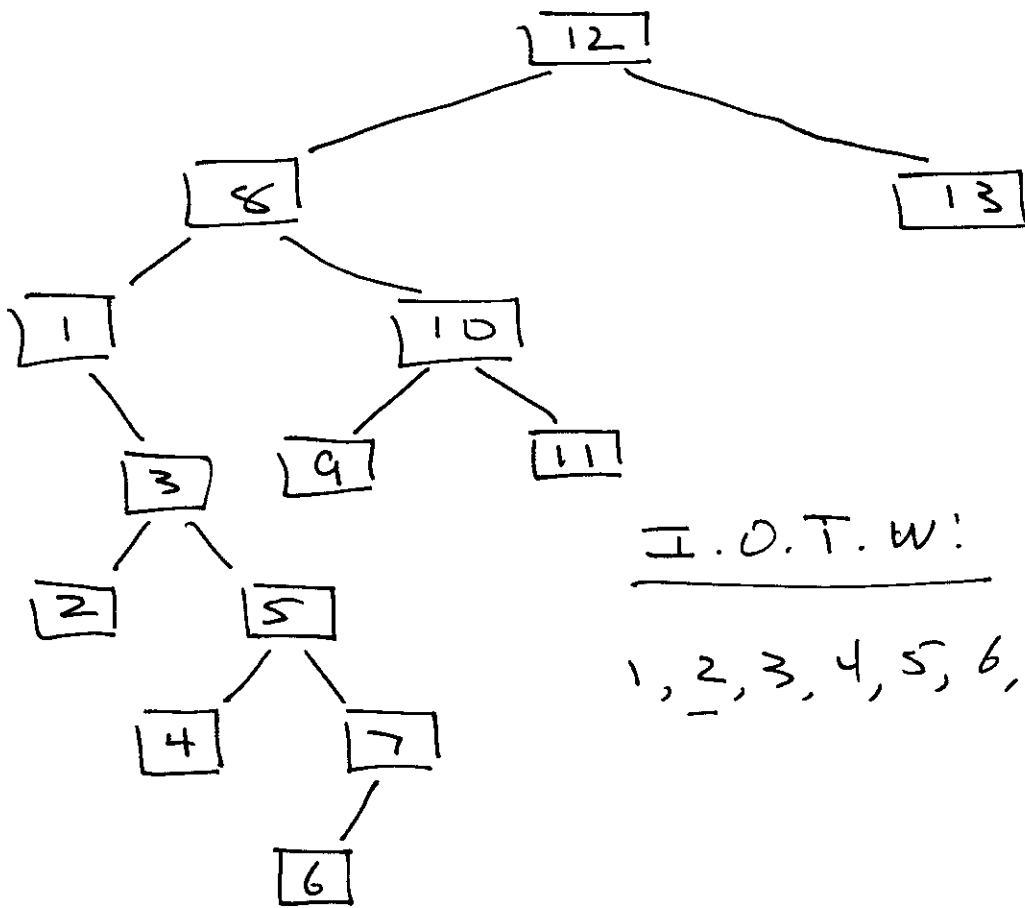
- $\text{TreeSearch}(x, k)$ ✓
- $\text{TreeMinimum}(x)$ ✓
- $\text{TreeMaximum}(x)$ ✓

Defn

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

Ex.

root

I.O.T.W!

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

Successor(1) = 2 (case 1)

Successor(8) = 9 (case 1)

Successor(7) = 8 (case 2)

have 2 cases for :

18

- TreeSuccessor(x)

case 1: x has a right child

successor of x is the leftmost descendant of that child.

case 2: x has no right child.

successor of x is lowest ancestor of x whose left child is also an ancestor of x . (note: x is its own ancestor)

Exercise

- define Predecessor of x
- write algorithm for TreePredecessor.