CS 61B Spring 2026

BSTs, 2-3 Trees, LLRBs

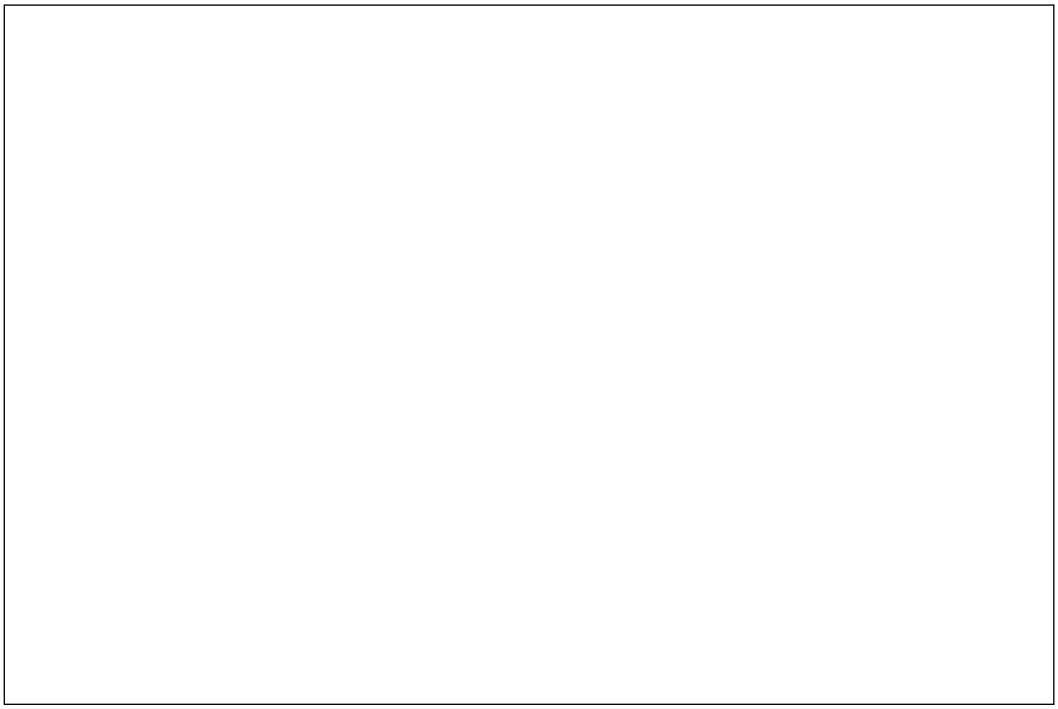
Discussion 08: March 9, 2026

1 Binary Search Trees

(a) We implement a binary search tree with the following methods:

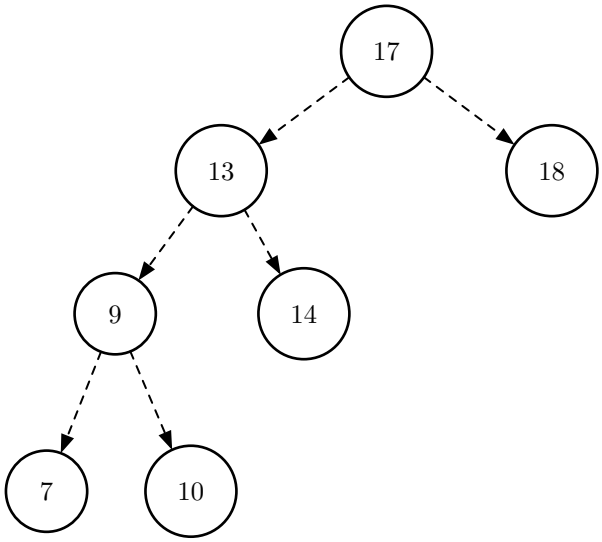
```
// Inserts an item into the binary search tree.
public void insert(T item) {
    // Implementation has been omitted
}

// Deletes an item from the binary search tree.
public void delete(T item) {
    // Implementation has been omitted
}
```

Draw the binary search tree that results from the following operations. Assume we start from an empty tree.



```
insert(5)
insert(7)
insert(10)
insert(6)
insert(3)
insert(1)
insert(4)
delete(10)
delete(7)
```

(b) Given the following binary search tree:

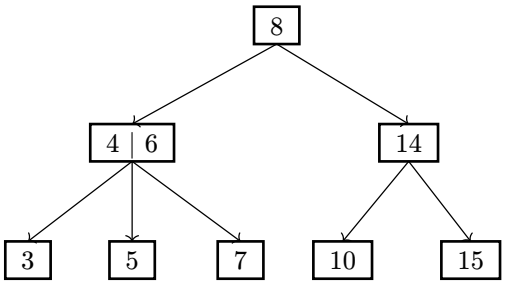


Suppose we delete the root node. Which node(s) can we replace 17 with as the new root node?

(c) Suppose we create a BST by inserting the nodes $V_0, V_1, ..., V_n$, where V_i is strictly smaller than $V_{\{i+1\}}$, in order. That is, we first insert V_0, then V_1, and so on. What is the runtime to find an element in this BST in the worst case, where N is the number of nodes?

2 2-3 Trees and LLRBs

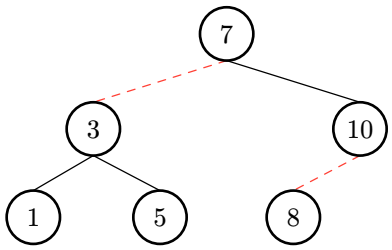
(a) Draw what the following 2-3 tree would look like after inserting 18, 38, 12, 13, and 20.



(b) Now, convert the resulting 2-3 tree to a left-leaning red-black tree.

(c) Suppose we create a 2-3 tree by inserting the nodes $V_0, V_1, ..., V_n$, where V_i is strictly smaller than $V_{\{i+1\}}$, in order. That is, we first insert V_0, then V_1, and so on. What is the runtime to find an element in this 2-3 tree in the **worst case**, where N is the number of nodes?

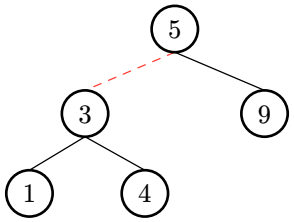
(d) Now, insert 9 into the LLRB Tree below. Describe where you would insert this node, and what balancing operations (`rotateLeft`, `rotateRight`, `colorSwap`) you'd take to balance the tree after insertion. Assume that in the given LLRB, dotted links between nodes are red and solid links between nodes are black.



(e) If a 2-3 tree has depth `H` (that is, the leaves are at distance `H` from the root), what is the maximum number of comparisons done in the corresponding red-black tree to find whether a certain key is present in the tree?

3 LLRB Insertions

Given the LLRB below, perform the following insertions and draw the final state of the LLRB. In addition, for each insertion, write the balancing operations needed in the correct order (`rotateRight`, `rotateLeft`, or `colorFlip`). If no balancing operations are needed, write "Nothing". Assume that the link between 5 and 3 is red and all other links are black at the start.



(a) 1. Insert 7

2. Insert 6

3. Insert 2

4. Insert 8

5. Insert 8.5

6. Final state

(b) Convert the final LLRB to its corresponding 2-3 Tree.