

Introduction to Data Structures

G. Philip

Institute of Mathematical Sciences, Chennai

gphilip@imsc.res.in

January 3, 2009

Designing a data structure

For the search-and-insert problem

- Input: A sorted list of n numbers, and a number x .
- Task: Find if x is in this list. If it is not, insert it in the proper position.

Example

- Is 20 present in $\langle 7, 11, 14, 15, 18, 26, 39, 50 \rangle$? If not, insert 20 in the proper position.



Designing a data structure

For the search-and-insert problem

- Input: A sorted list of n numbers, and a number x .
- Task: Find if x is in this list. If it is not, insert it in the proper position.

Example

- Is 20 present in $\langle 7, 11, 14, 15, 18, 26, 39, 50 \rangle$? If not, insert 20 in the proper position.

Question:

- What is a good way to represent this list of numbers?



Designing a data structure

- Goal: Try to design a data structure that has the good properties of arrays and linked lists:
 - Finding an element should take $O(\log n)$ time.
 - Inserting an element should take constant time.



Designing a data structure

- Goal: Try to design a data structure that has the good properties of arrays and linked lists:
 - Finding an element should take $O(\log n)$ time.
 - Inserting an element should take constant time.
- We may or may not succeed in designing such a data structure, but it is good to try...



Designing a data structure

Question:

- Why does insertion into a linked list take only constant time?

Hint:

- Contrast with arrays, where it could take much more time...



Designing a data structure

Question:

- Why does insertion into a linked list take only constant time?

Hint:

- Contrast with arrays, where it could take much more time...

Design decision:

- We will use a linked structure, not an indexed structure like an array.



Designing a data structure

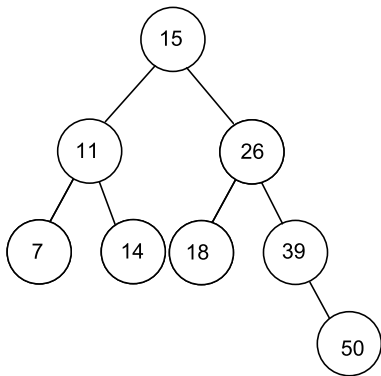
Question:

- Why is it possible to search in $O(\log n)$ time in a sorted array?



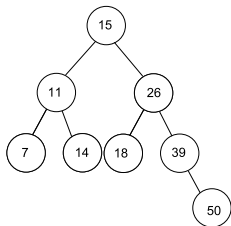
Designing a data structure

Binary search tree



Designing a data structure

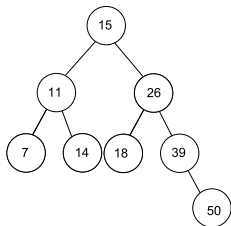
Binary search tree



- A binary search tree is
 - A binary tree.
 - Has a distinct value stored at each node.
 - Let val be the value stored at a node. Then:
 - The values in the subtree rooted at the left child of the node are all less than or equal to val .
 - The values in the subtree rooted at the right child of the node are all greater than or equal to val .

Designing a data structure

Binary search tree

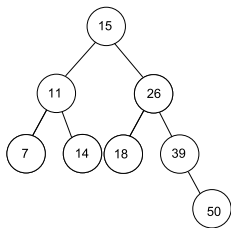


Problem:

- Find if 20 is present in this tree.

Designing a data structure

Binary search tree

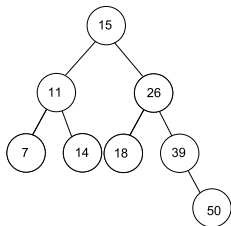


Problem:

- Find if 20 is present in this tree.
- How much time does this take?

Designing a data structure

Binary search tree

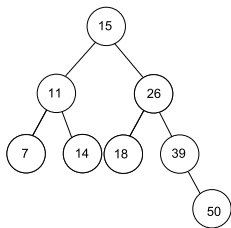


Problem:

- Insert 20 in the correct position in this tree.

Designing a data structure

Binary search tree



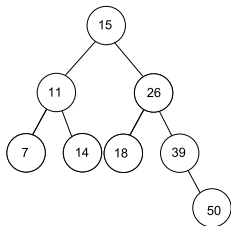
Problem:

- Insert 20 in the correct position in this tree.
- How much time does this take?



Designing a data structure

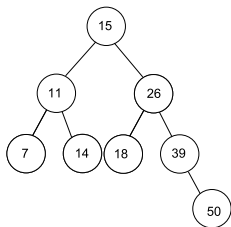
Binary search tree



- We can find if a number is present in a binary search tree in $O(\log n)$ time.
- We can insert a missing number into a binary search tree in constant time.

Designing a data structure

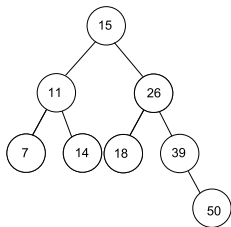
Binary search tree



- We can find if a number is present in a binary search tree in $O(\log n)$ time.
- We can insert a missing number into a binary search tree in constant time.
- Problem solved!

Designing a data structure

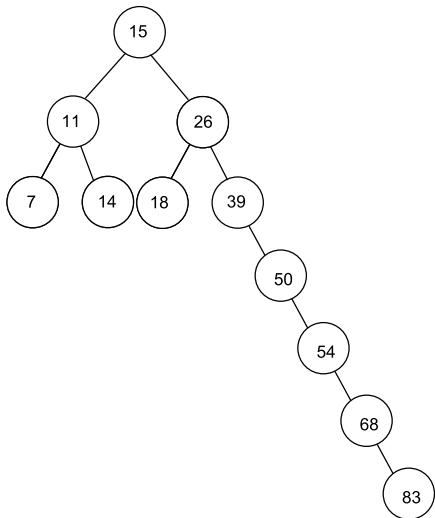
Binary search tree



- We can find if a number is present in a binary search tree in $O(\log n)$ time.
- We can insert a missing number into a binary search tree in constant time.
- Problem solved!
- But...

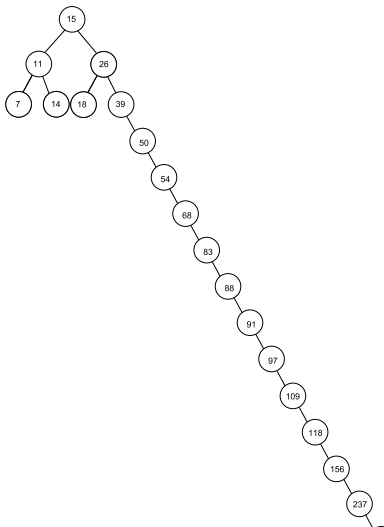
Binary Search Tree

The problem



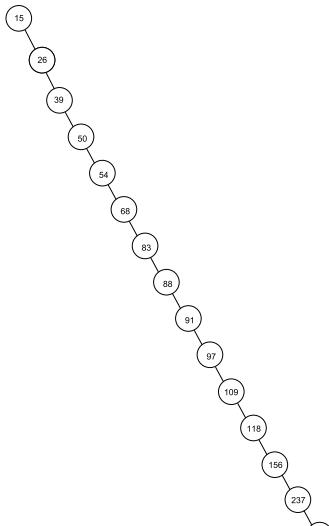
Binary Search Tree

The problem



Binary Search Tree

The problem



Binary Search Tree

The problem

- The tree can become skewed, or unbalanced.
- In the worst case, this can lead to...



Binary Search Tree

The problem

- The tree can become skewed, or unbalanced.
- In the worst case, this can lead to...
- $O(n)$ time for searching – as bad as a linked list!



Problem: a binary search tree can become skewed

Solutions

- The tree can become skewed, resulting in $O(n)$ search times.
- Solutions for this problem use the "rotation" operation.



Binary search trees

Tree rotation

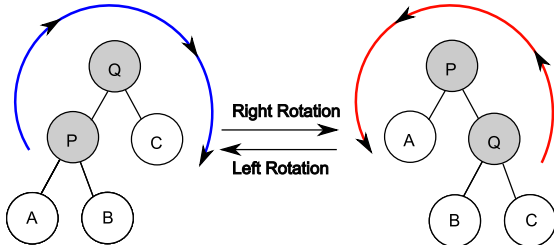
- Changes the structure of a BST without messing up the BST ordering.
- Moves one node up and one node down, and moves the subtrees of these nodes preserving the ordering.
- Changes the shape of the tree – reduces its height.



Binary search trees

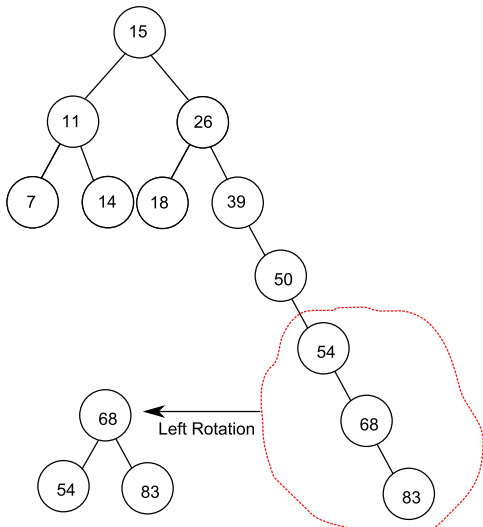
Tree rotation

- Changes the structure of a BST without messing up the BST ordering.
- Moves one node up and one node down, and moves the subtrees of these nodes preserving the ordering.
- Changes the shape of the tree – reduces its height.



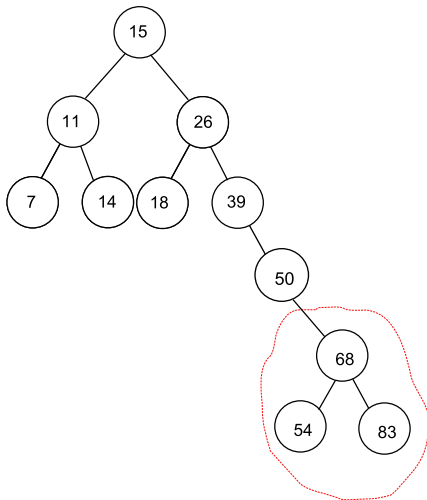
Binary search trees

Tree rotation



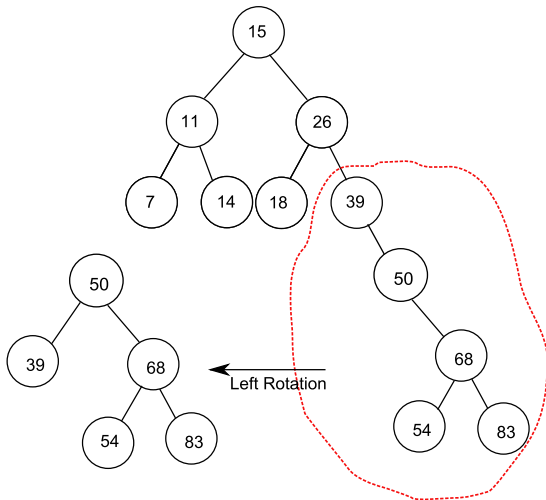
Binary search trees

Tree rotation



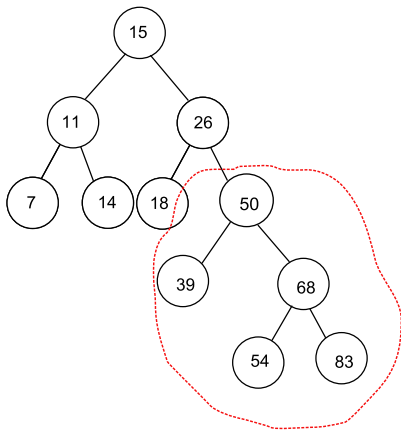
Binary search trees

Tree rotation



Binary search trees

Tree rotation



Balancing binary search trees

- Many methods that keep a binary search tree balanced, using tree rotation:
 - AVL trees
 - Red-Black trees
 - Splay trees



Questions?

