

CS2010 Semester 1 2012/2013
Data Structures and Algorithms II

Tutorial 02 - Binary Search Trees

For Week 04 (03 September - 07 September 2012)

Released: Wednesday, August 29, 2012

Document is last modified on: July 27, 2012

1 Introduction and Objective

The purpose of this tutorial is to reinforce the concepts of Binary Search Tree (BST) and the importance of having balanced BST. We will also discuss Subtask 1 of PS1 during this tutorial.

2 Tutorial 02 Questions

Operations on BSTs

Q1. Find the successor of 33 in the BST (and show how to find it)

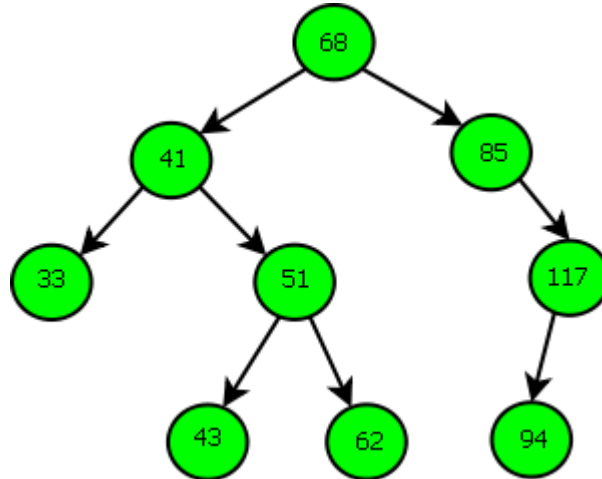


Figure 1: BST for Q1 and Q2

1. 68
2. 43
3. 85
4. 41
5. 62

Q2. Find the predecessor of 94 in the same BST in Figure 1 (and show how to find it)

1. 17
2. 41
3. 68
4. 33
5. 85

Q3. What sequence does a preorder traversal of the BST in Figure 3 yield?

Preorder traversal is very similar to Inorder traversal that we have seen earlier in Lecture 2.

Preorder traversal is just like this:

```

PreOrder(T)
  if T is null, stop
  Visit/Process T (see, the visitation is done first)
  PreOrder(T.left)
  PreOrder(T.right)

```

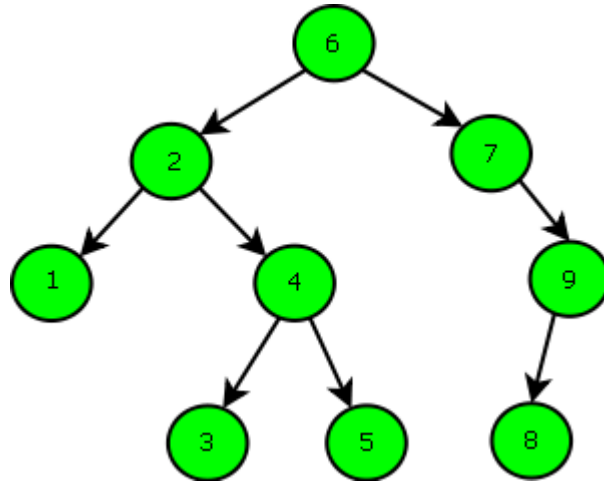


Figure 2: BST for Q3

1. 1, 2, 3, 4, 5, 6, 7, 8, 9
2. 9, 8, 7, 6, 5, 4, 3, 2, 1
3. 1, 3, 5, 4, 2, 8, 9, 7, 6
4. 6, 2, 1, 4, 3, 5, 7, 9, 8

Q4. Can you re-write the successor algorithm without using the pointer comparison?
(i.e without the use of `(cur == par.right)`)

Note: You can do the same for predecessor algorithm.

Q5. Is it possible to enforce that a BST is always a full binary tree?

Full binary tree is defined as a tree where each node other than the leaf has exactly two children.

AVL Trees

Q6. Is the following a valid AVL tree?

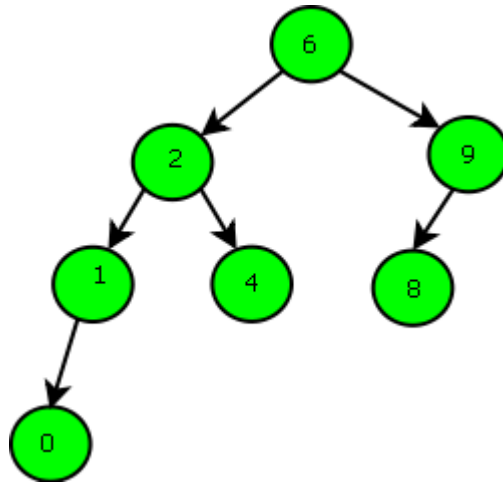


Figure 3: AVL Tree for Q6

Q7. Show the sequence of rotations required to re-balance the following AVL tree if 77 is inserted. (Indicate the balancing factor at each step).

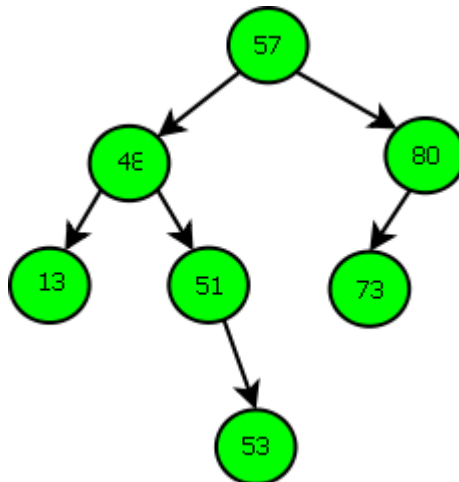


Figure 4: AVL Tree for Q7

Problem Set 1

Q8. Discussion of PS1 subtask 1.