Week 2

ASSIGNMENTS

~~**h0** available and due before 10pm on Monday 1/28~~ - LAST NIGHT

→ **h1** available and due before 10pm on Monday 2/4

→ **p1** available and due before 10pm on Thursday 2/7

→ **TA Lab Consulting** - See schedule (cs400 home pages) - TestDS_My

→ **Peer Mentoring available** - Friday 8am-12pm, 12:15-1:30pm in 1289CS

In-class Demo
Open Eclipse and
Web Browser to p1 assignment.

→ **TAs and Peer Mentors will focus on helping students get p1 JUnit tests running in Lab.**

→ **We can not guarantee that we can get your personal computers configured.**

Module: Week 2 (start on week 3 before next week)

THIS WEEK

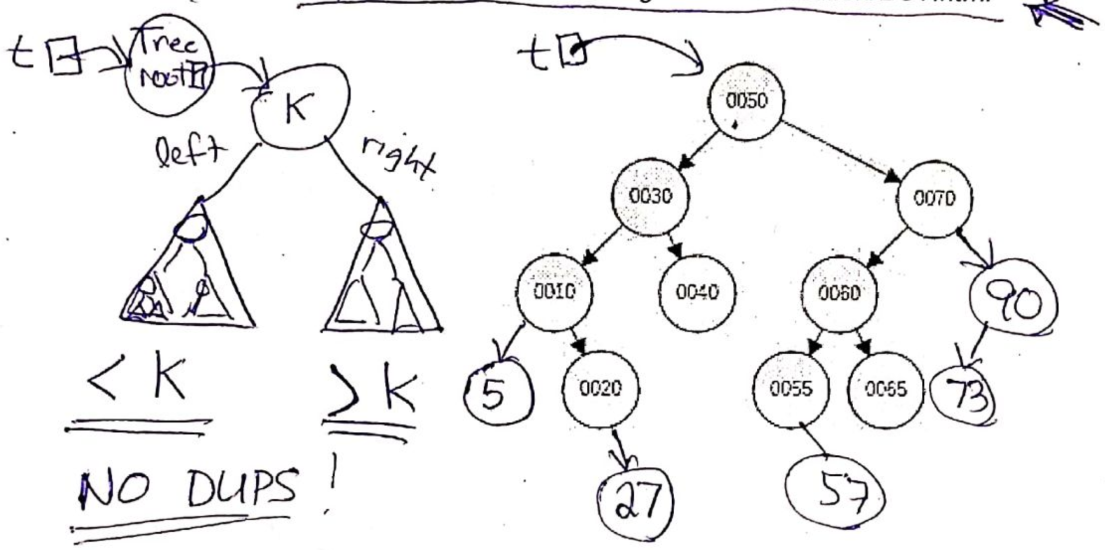
- → Ready Set Program 1!
  - Read Assignment - there are getting started instructions there
  - Create and configure project for JUnit5, compile and run **TestDS_My**
  - Put tests in DataStructureADTTest (not the sub-classes)
- → Testing: JUnit5
- → Java: inner classes
- Determining Height of a Tree (Recursion Review) p1 demo
- → Binary Search Trees (BST) (Review?)
  - operations
  - implementing
  - complexities
- Classifying Binary Trees
- Balanced Search Trees
- → George Adelson-Velsky and Evgenii Landis

NEXT WEEK

- X-team Exercise x1 (in-class exercise with your assigned teams)
- Watch for instructions to find your team number and how to meet

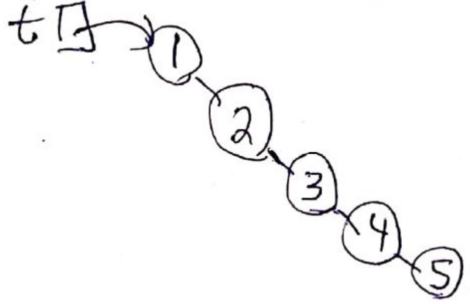
Binary Search Tree (BST) Review

Image created at: https://www.cs.usfca.edu/~galles/visualization/BST.html



Insert 5, 27, 90, 73, 57 into the above BST tree (recall binary search algorithm)

Insert the values 1,2,3,4,5 into an empty BST



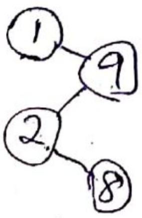
What can you conclude about the shape of a BST when the values are inserted in sorted order?

It's linear. O(N)

FATAL FLAW
- Shape depends upon insert order
- Complexity depends upon shape

Will you only get this shape if inserted in sorted order?

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

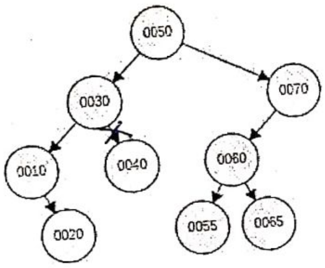


Deb

Practice Deleting from a BST

Delete 90 from this tree.
Delete 40 and then 65 from this tree.

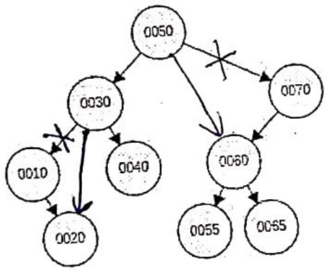
unlink 40
By setting right-child
of 30 to null



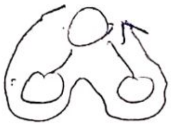
unlink 65
set right-child
of 60 to null

Delete 10 and then 70 from this tree.

delete (10)
set left-child
of 30
to 20 (right-child of 10)

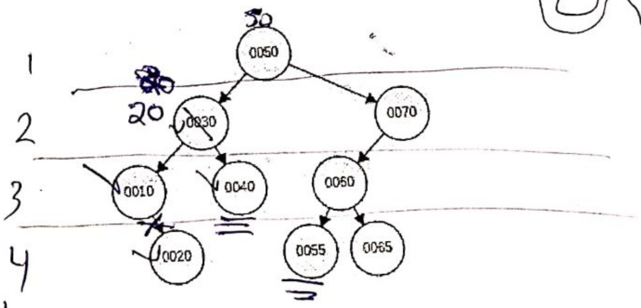


delete (70)
set right-child
of 50 (root)
to 60 (left-child of 70)



How do you delete 30 or 50 from this tree?

Delete 30
(?)



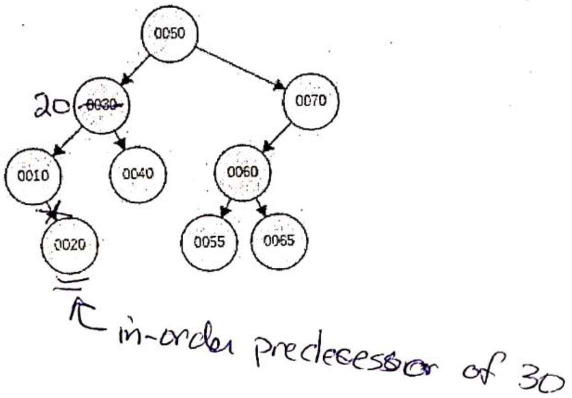
in-order traversal

in 10, 20, 30, 40, 50, 55, 60, 65, 70 —— LVR in-order
pre-order 50, 30, 10, 20, 40, 70, 60, 55, 65 —— VLR pre-order
post-order 20, 10, 40, 30, 55, 65, 60, 70, 50 —— LRV post-order
level-order 50, 30, 70, 10, 40, 60, 20, 55, 65 — level

Deb

Delete 30 from this tree using the in-order predecessor

1. replace 30 with value of in-order predecessor

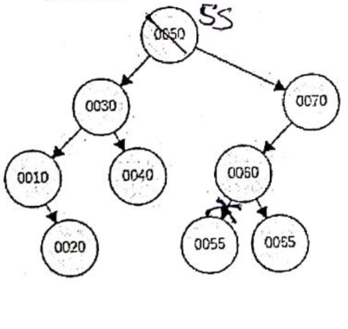


2. delete 20 from left subtree of 30

Delete 50 from this tree using the in-order successor

1. replace 50 with 55

2. delete 55 from right-subtree of 50



How do we find in-order predecessor or in-order successor?

furthest to right of left-subtree
greatest value in left-subtree
go left - then right as far as possible