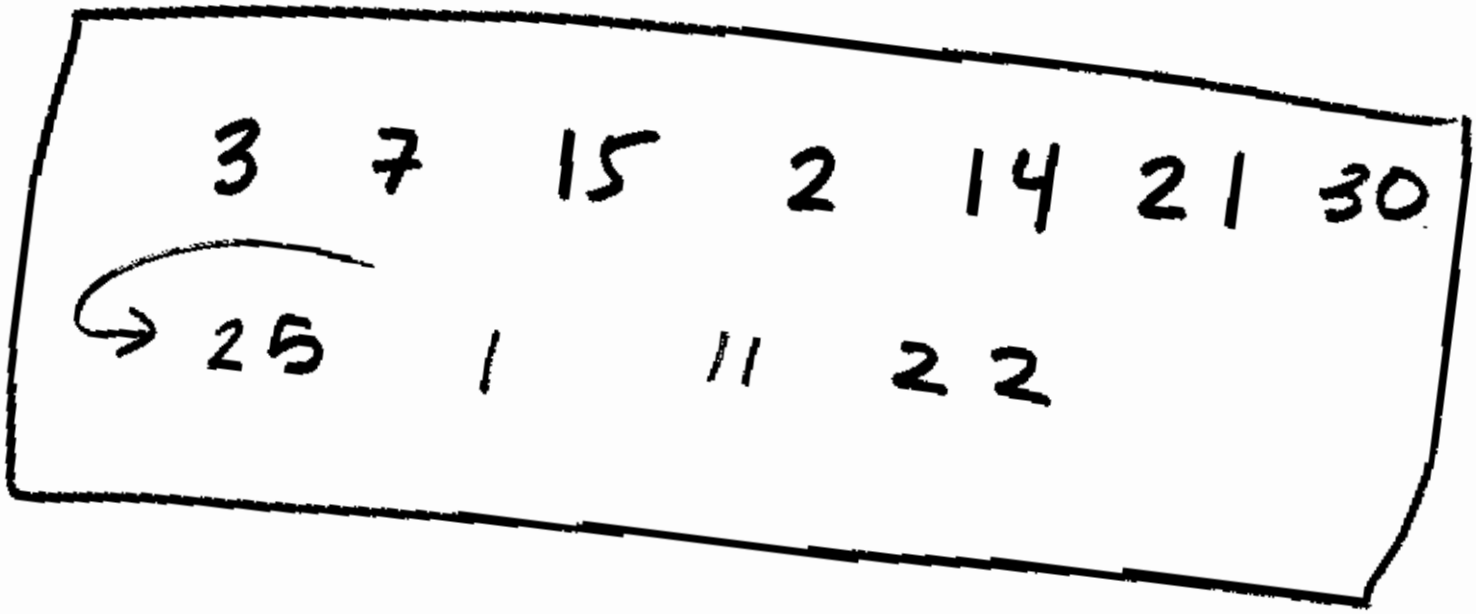
Assignment (ungraded)

2-3, 2-3-4, red black, AVL



OnLine Final Time

CH 269 June 9
(Tuesday)

[illegible] 19:30-2120

	May 14 Advanced Trees
May 19 Advanced trees & Efficiency	21 Graphs & heaps
May 26 Sorting	28 Sorting
June 2 Review	June 4 Review for Final
FINAL	

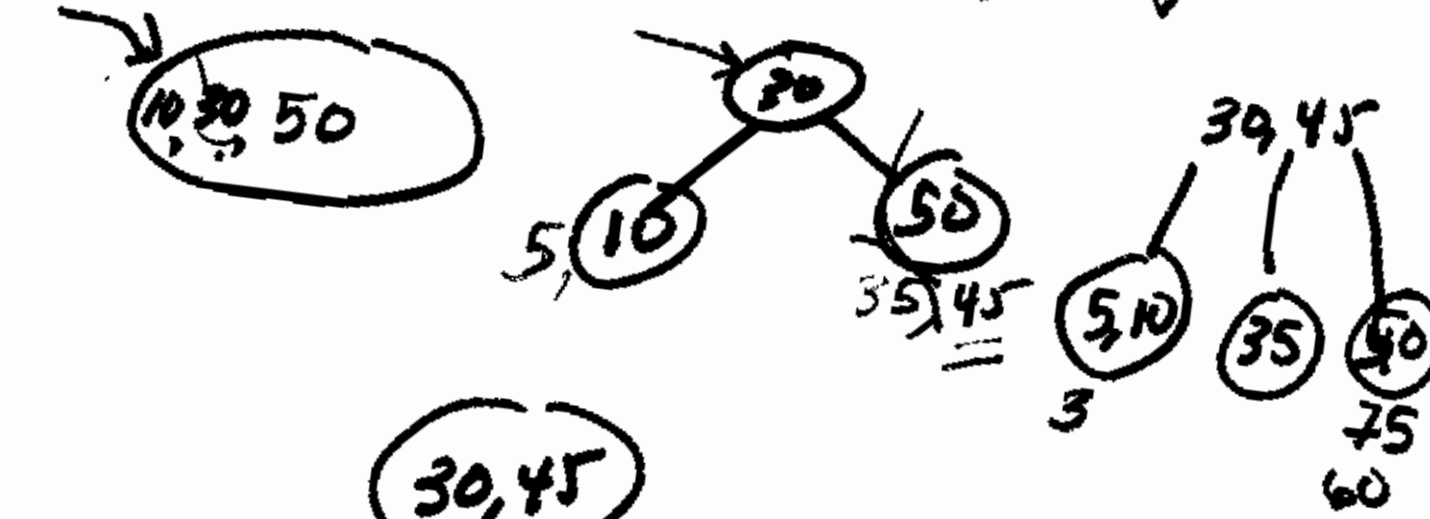
2-3-4 tree

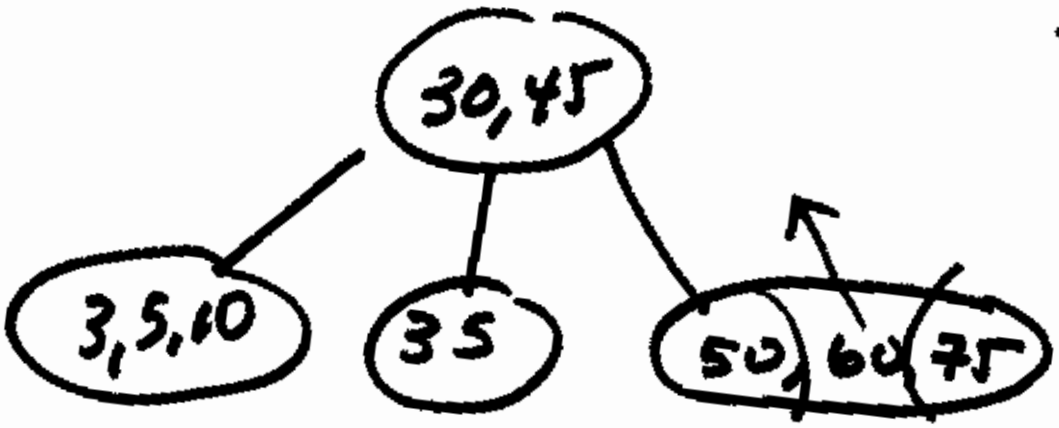
Algorithm for Insert

- Begin at Root
- Traverse down tree to a leaf going the correct direction based on binary search alg.
- IF we encounter a node with 3 data items as we travel down
  - Stop
  - Send the middle data ("push up") to it's parent
  - split the Node
- When we finally reach a leaf, add the data there.

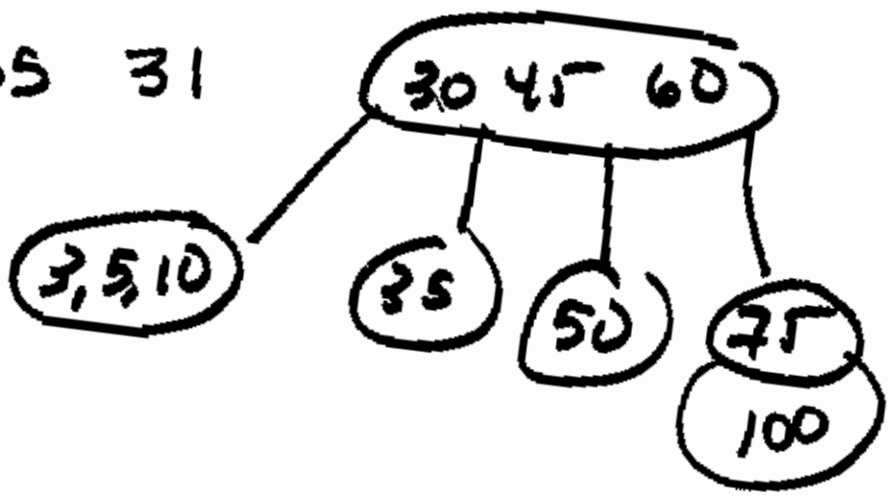
2-3-4

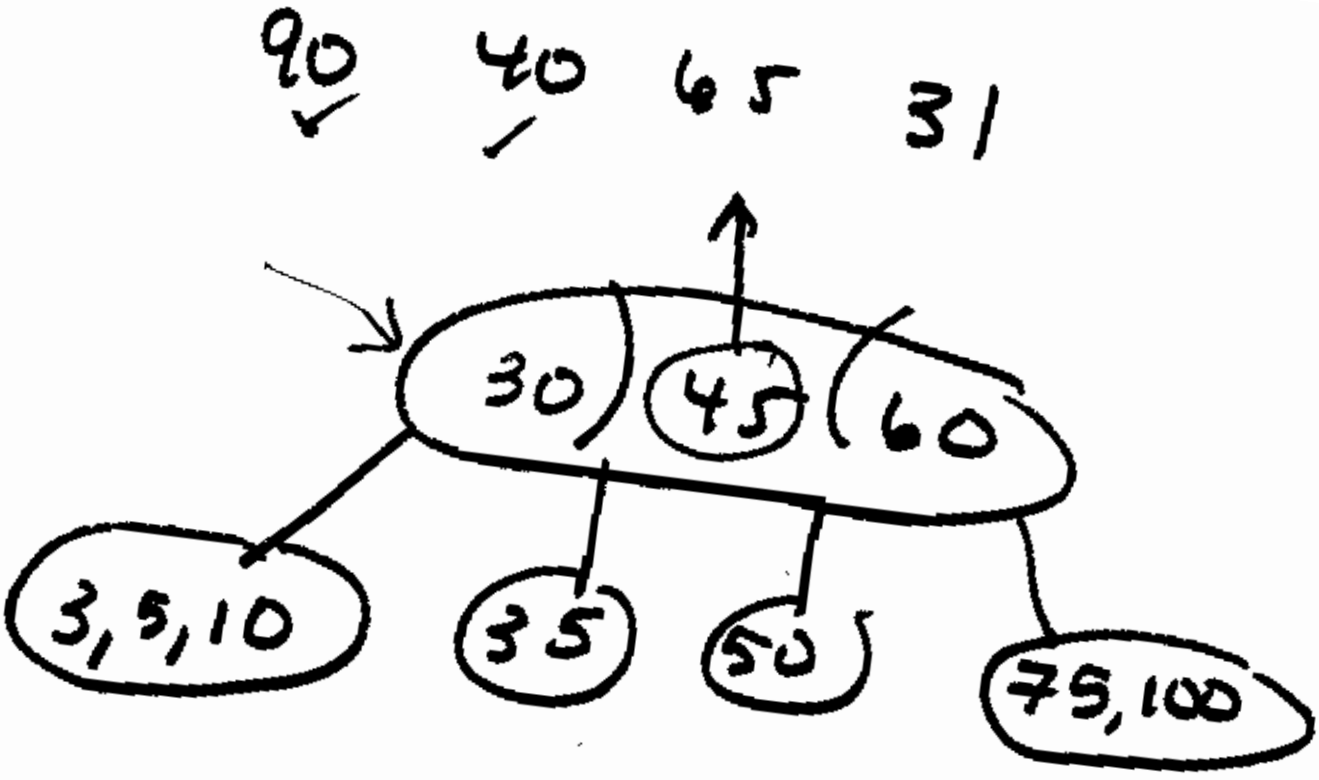
50 30 10 5 35 45 75 60 3

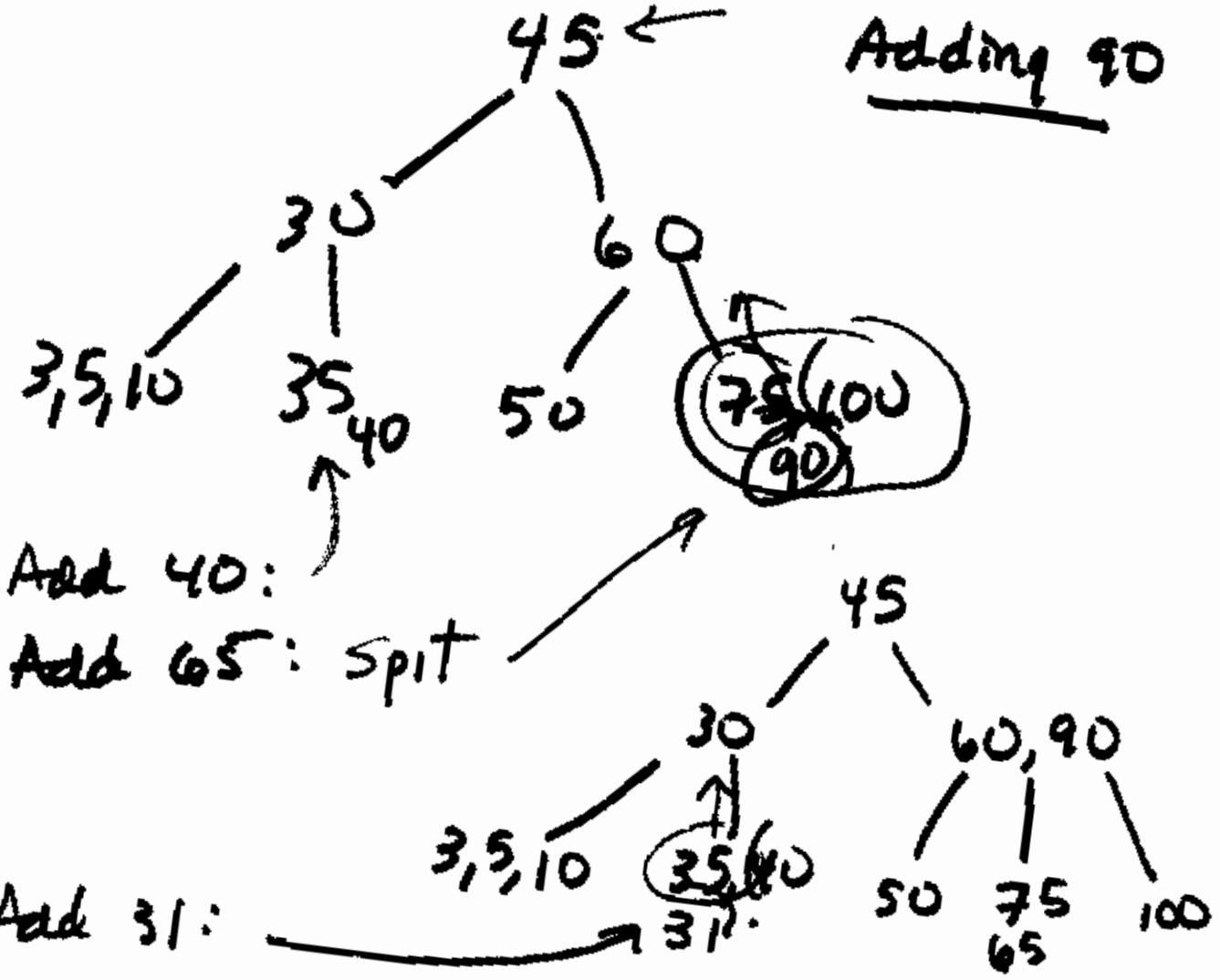




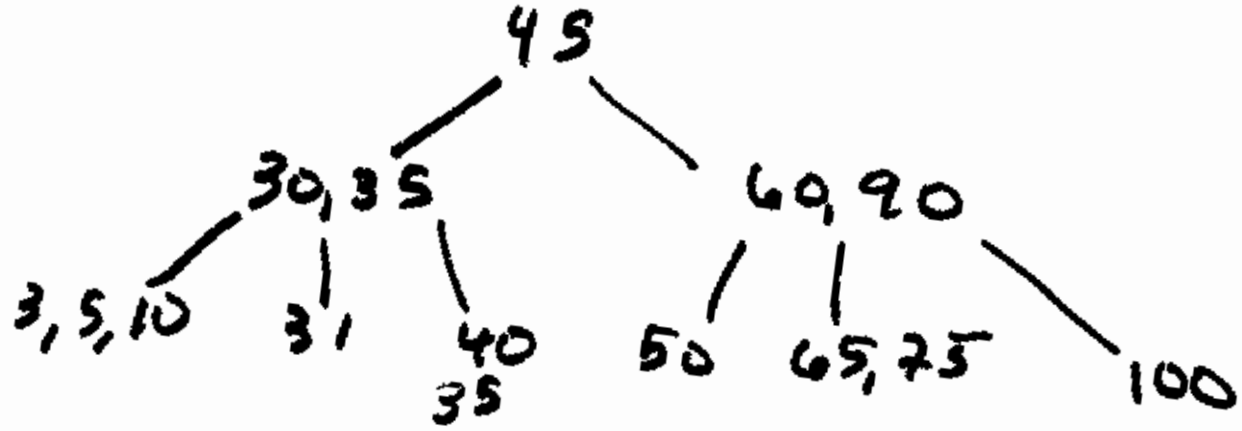
100 90 40 65 31




90
40
65
31
30
45
60
3,5,10
35
50
75,100

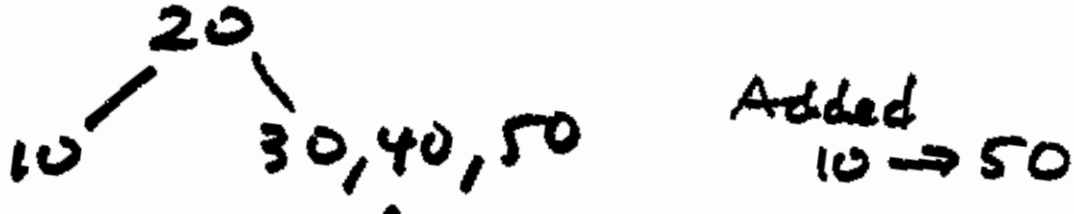

45
Adding 90
30
60
3,5,10
35
40
50
75
100
90
Add 40:
Add 65: split
45
30
60,90
3,5,10
35
40
50
75
65
100
Add 31:
31

Add 35 again ....



10, 20, 30, 40, 50, 60, 70, 80, 90, 100

(10, 20, 30)

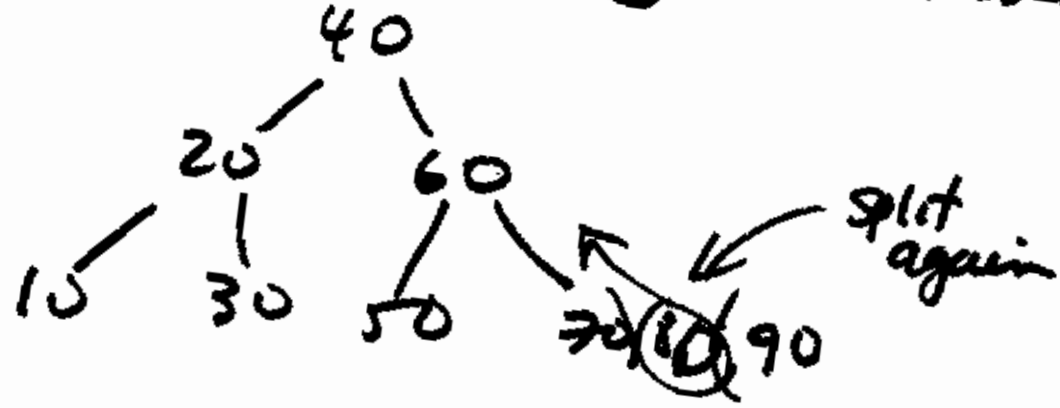


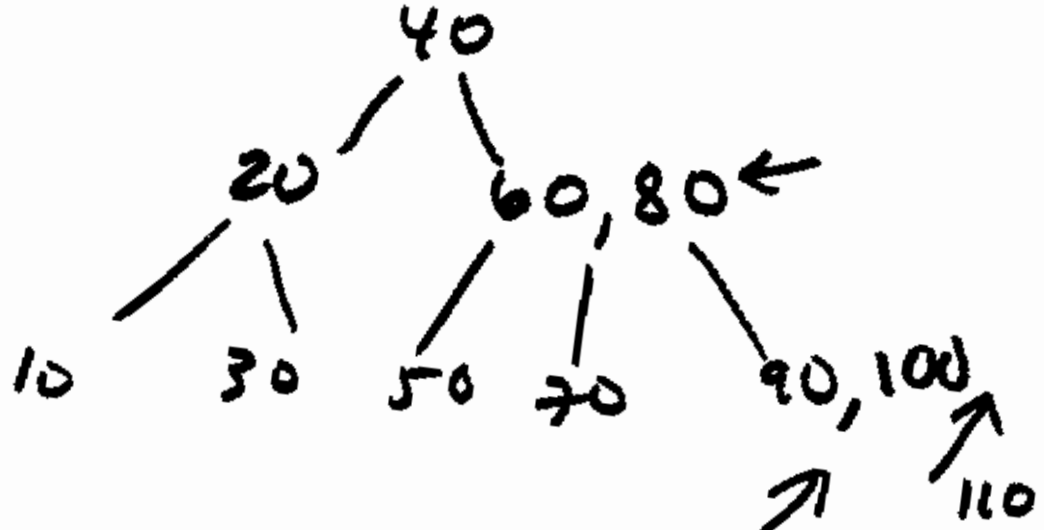
Added 10 ⇒ 50

Add 60: split

Add 80: split

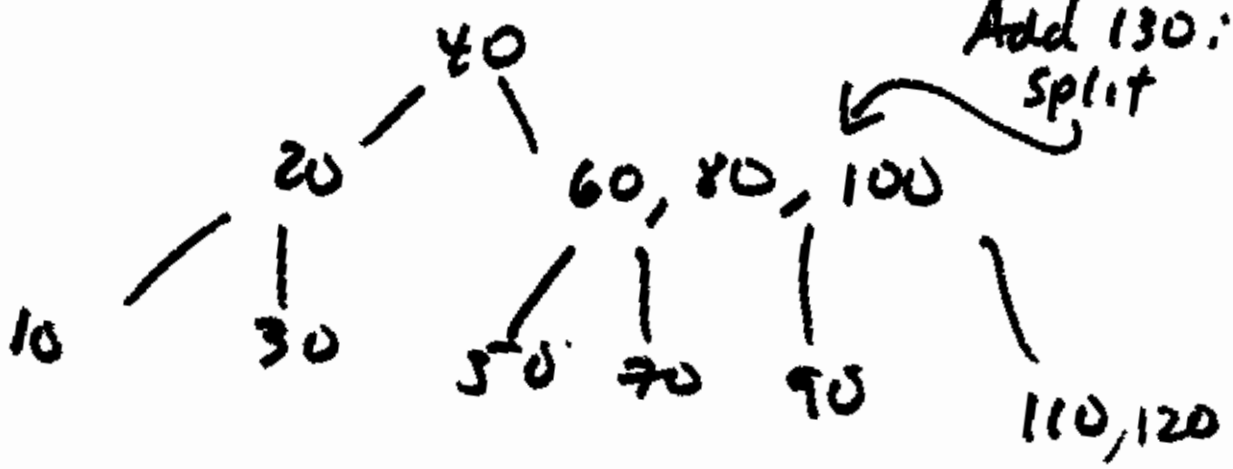
Add 100: Immediately split root node

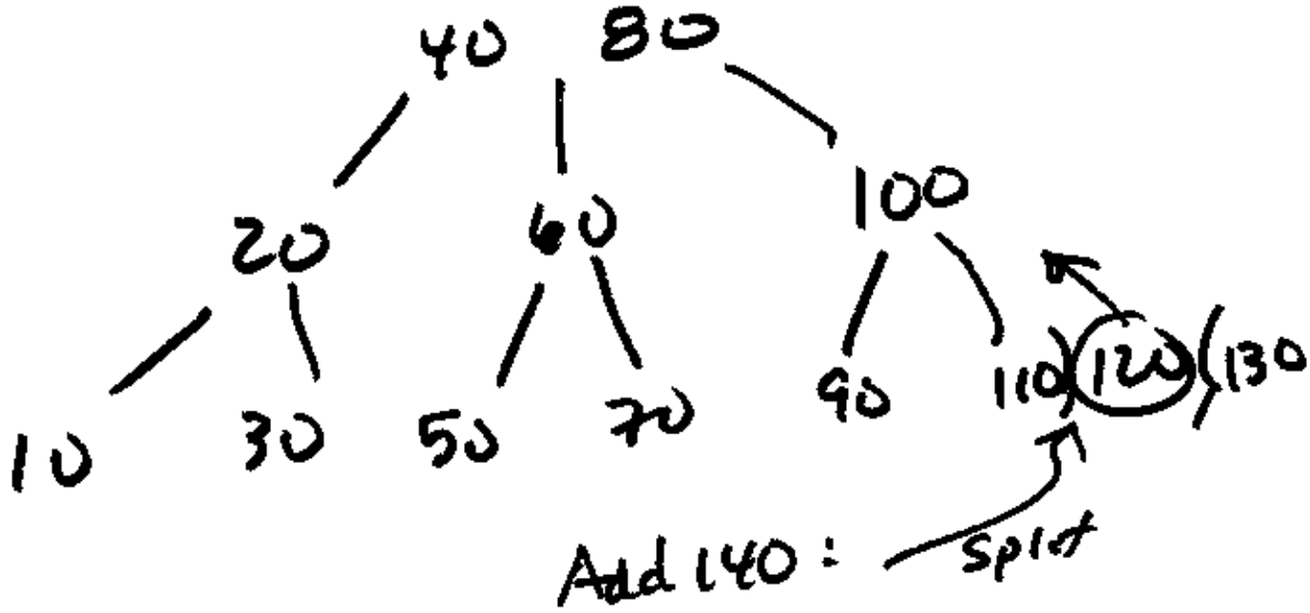


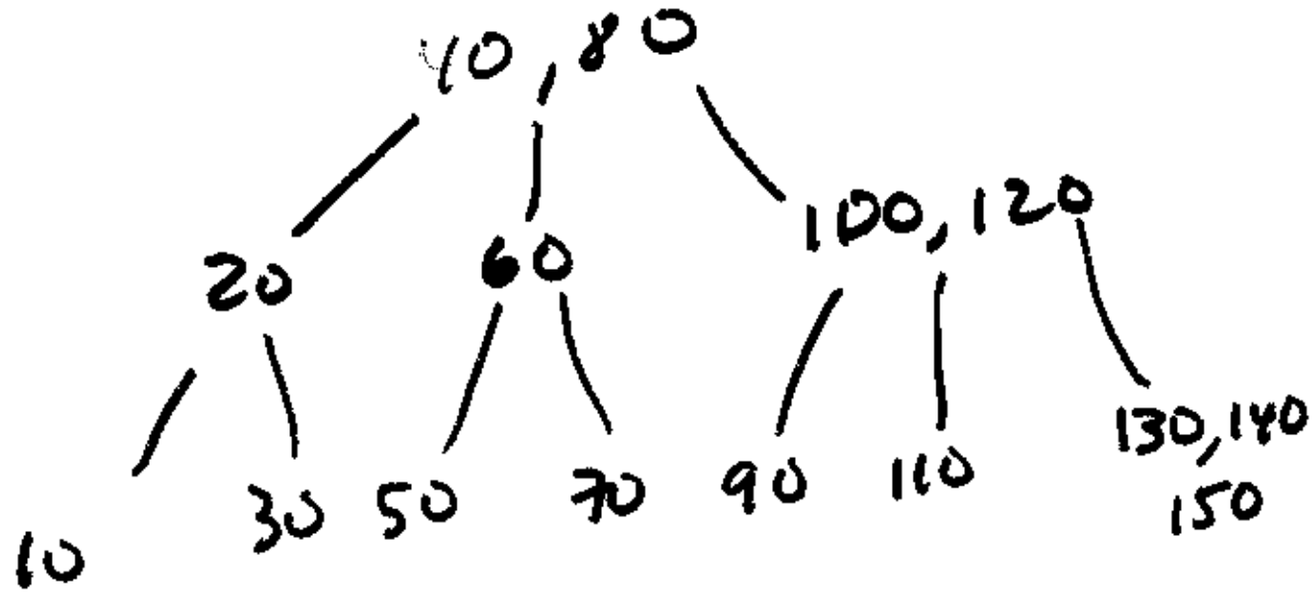


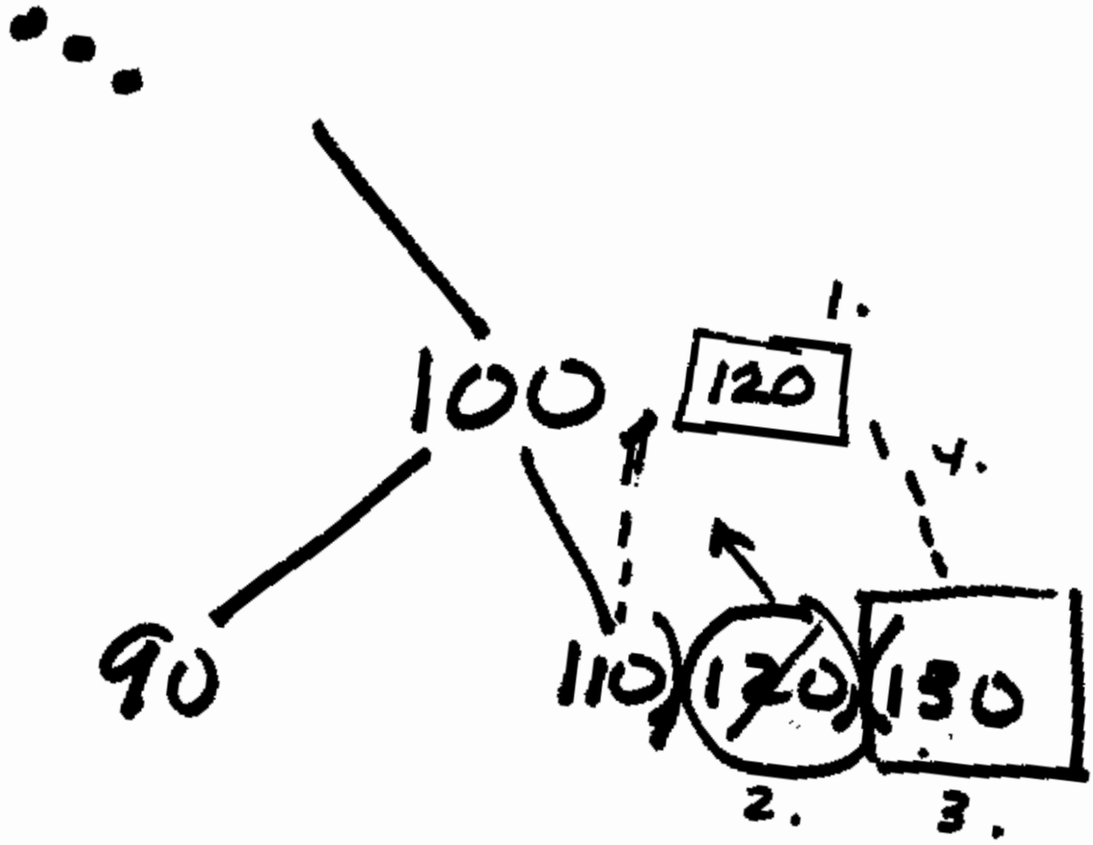
110, 120, 130, 140, 150

Add 120: split




40
80
20
60
100
10
30
50
70
90
110
120
130
Add 140 : split

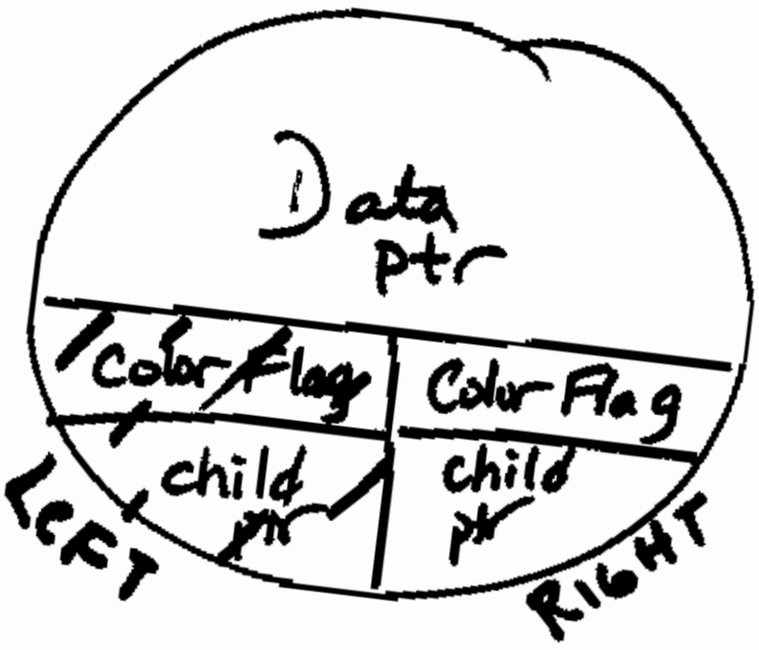

40, 80
20
60
100, 120
10
30
50
70
90
110
130, 140
150



~14 pointers

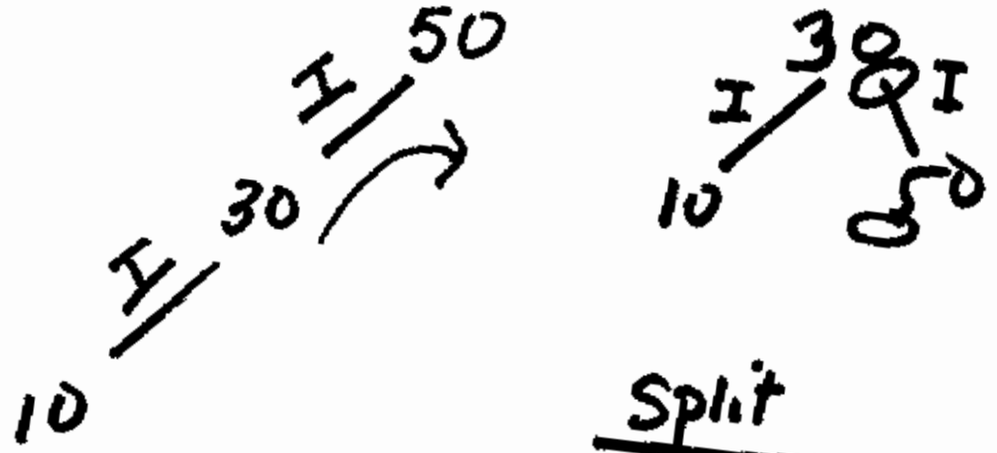
1. Copy ptr to 120
2. Clear out un-used pointers } 2 data, 2 child ptr
3. New Node for split
   - copy pointer
   - set other ptrs } 7 pointers
4. Copy the child ptrs in the parent (2)

Red Black

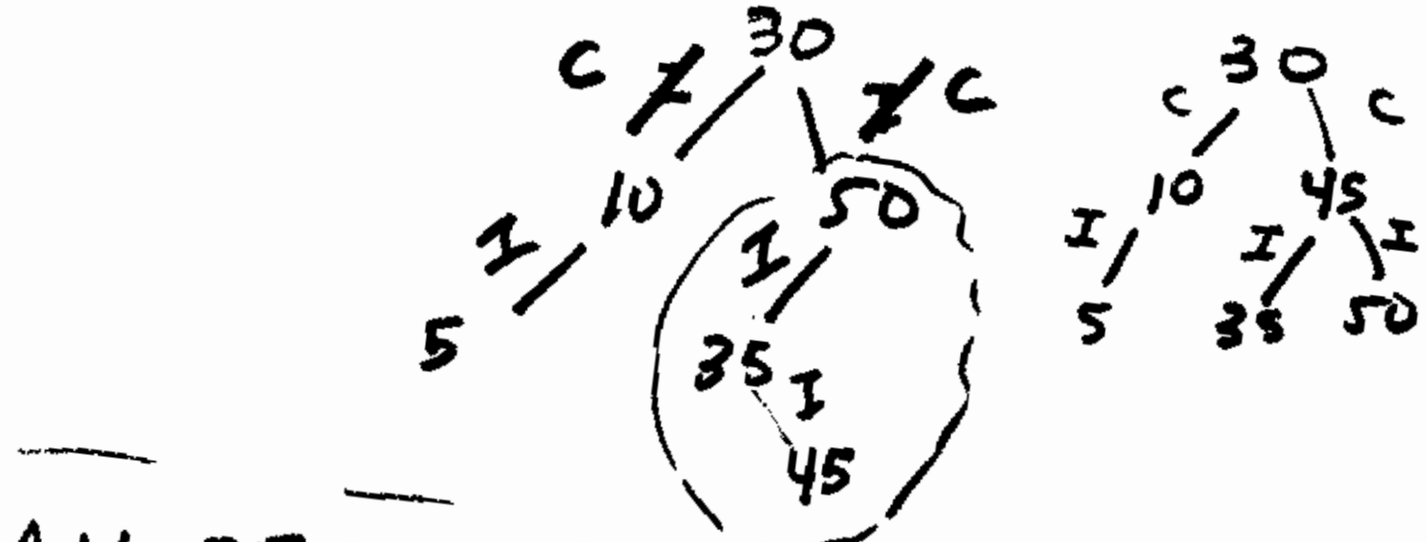

Data
ptr
Color Flag
Color Flag
child
ptr
child
ptr
LEFT
RIGHT

Red Black

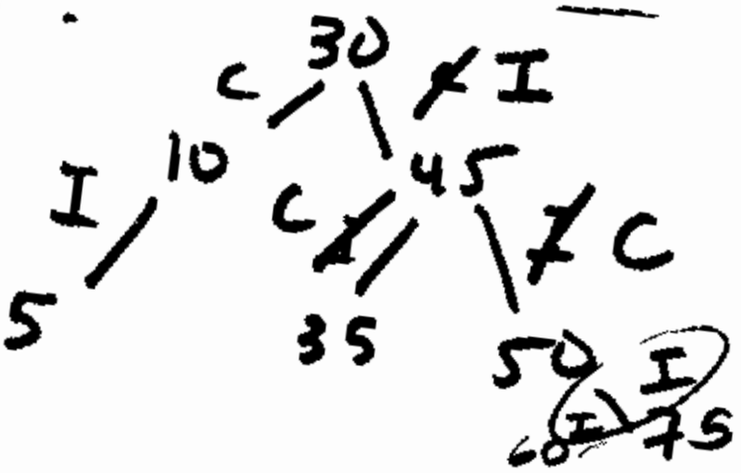
50 30 10 5 35 45 75 60



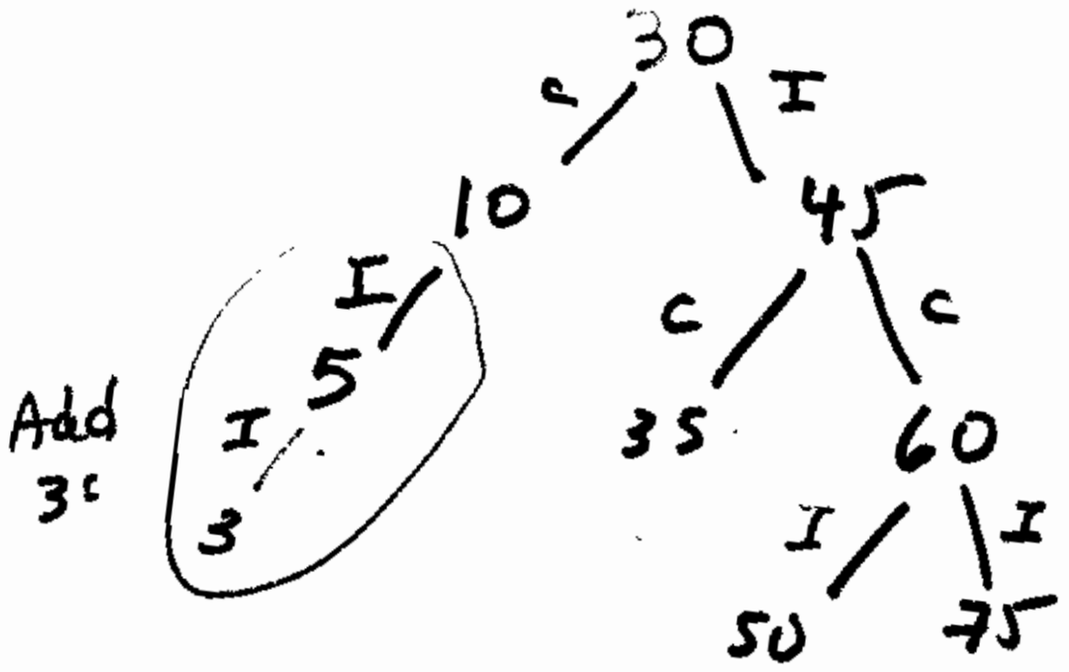
Split

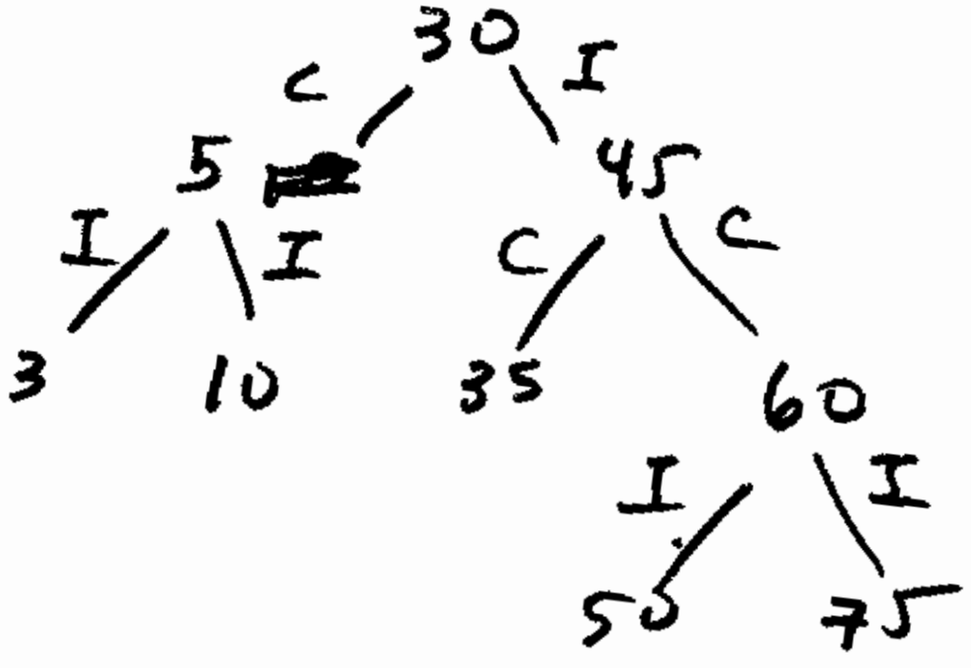


Add 75:



Add 60:

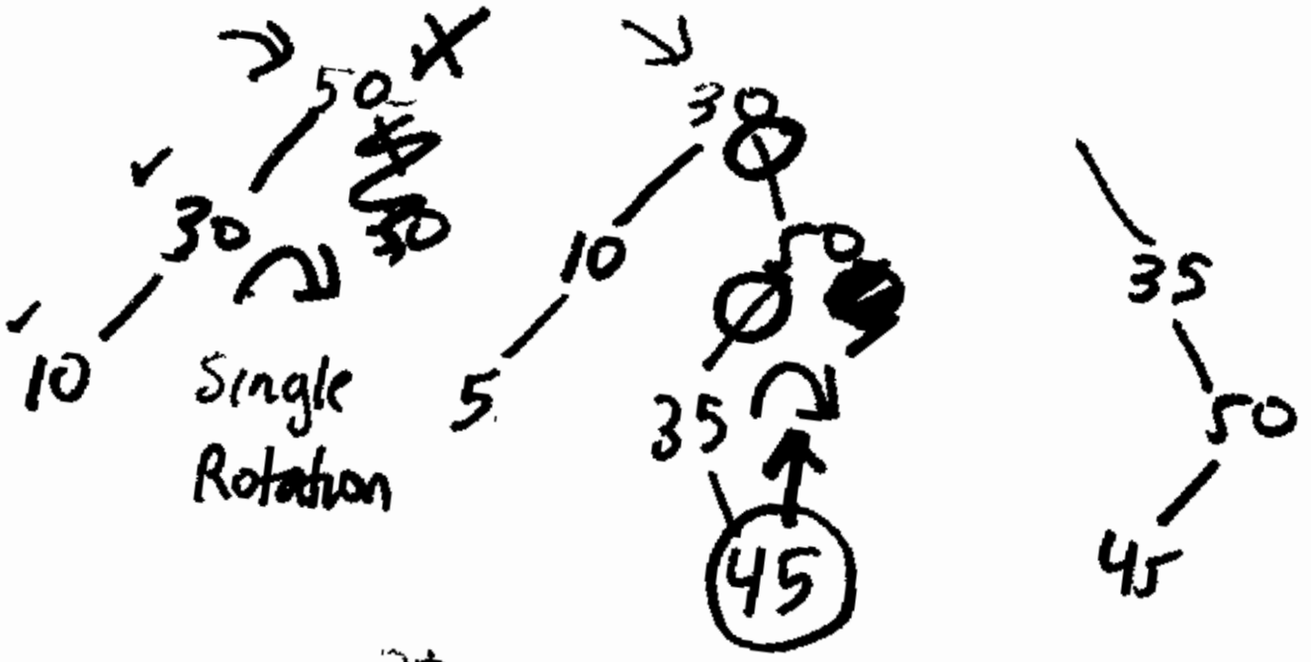

30
C
I
10
45
I
C
C
5
35
60
Add
3:
I
I
I
3
50
75

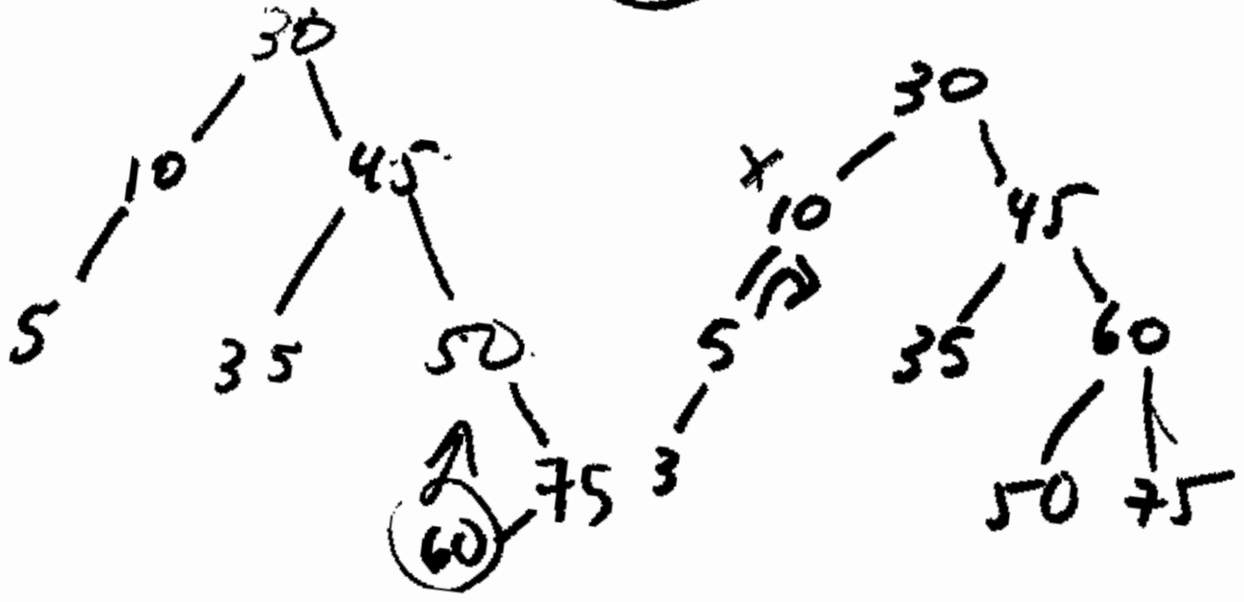

30
C
I
5
45
I
I
C
C
3
10
35
60
I
I
50
75

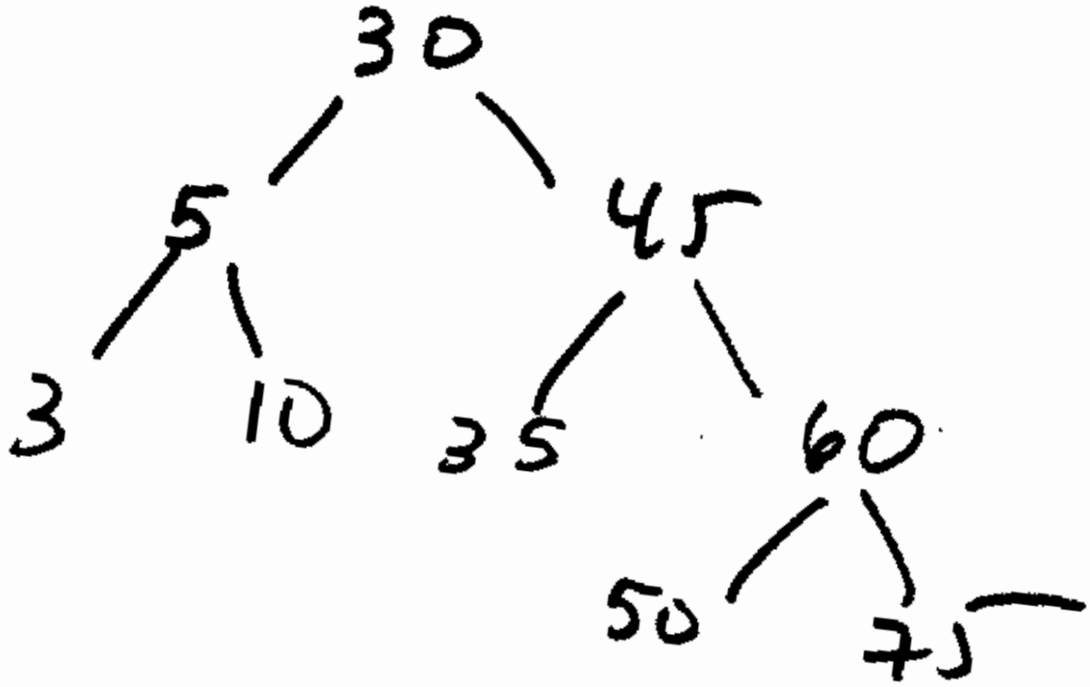
AVL

BST + Balance factors
or Height values

50 30 10 5 35 45 75 60






30
5
45
3
10
35
60
50
75