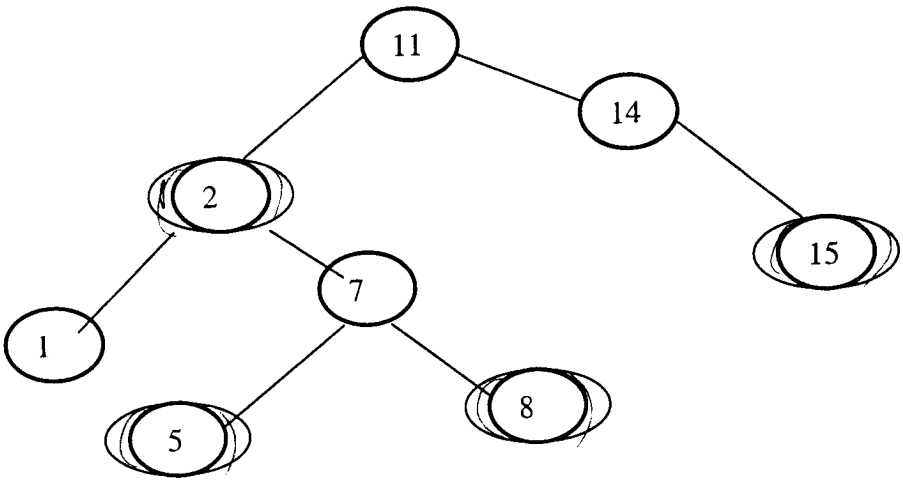
Notre Dame University
Faculty of Natural And Applied Sciences

CSC 313 - Test #2

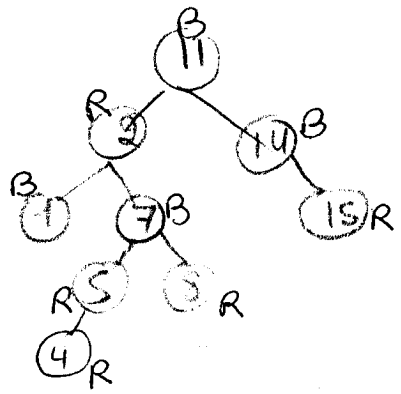
grade 93

Name: _____ _

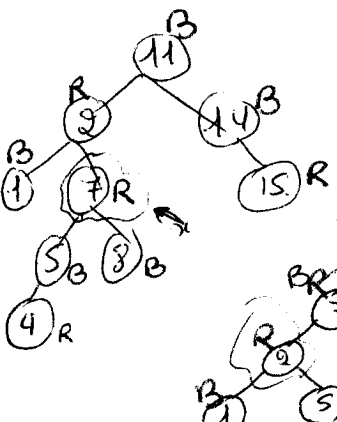
1. In the following binary search tree, let the simple circled nodes be black, and the double-circled nodes are red. **(15 points)**



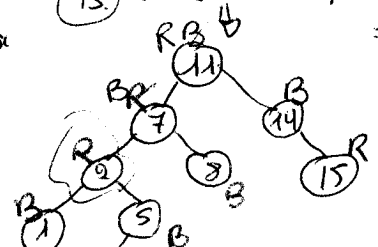
Insert 4



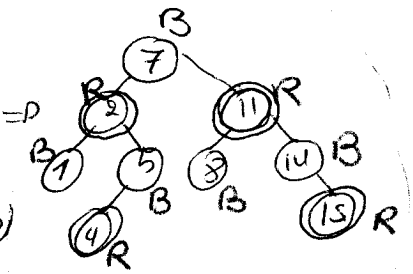
Violation at 4.
Red → Red Child!!
uncle Red (8)
⇒ case 1 : uncle & parent → Black
parent of par → Red



violation at 7
uncle Black, Right child
Case ②
{ x = p → x
Rotate left(x)



⇒ case 3 (left child)
⇒ p → p = Black
x → p → p = Red.
Right Rotate (x → p → p)
⇒ 11



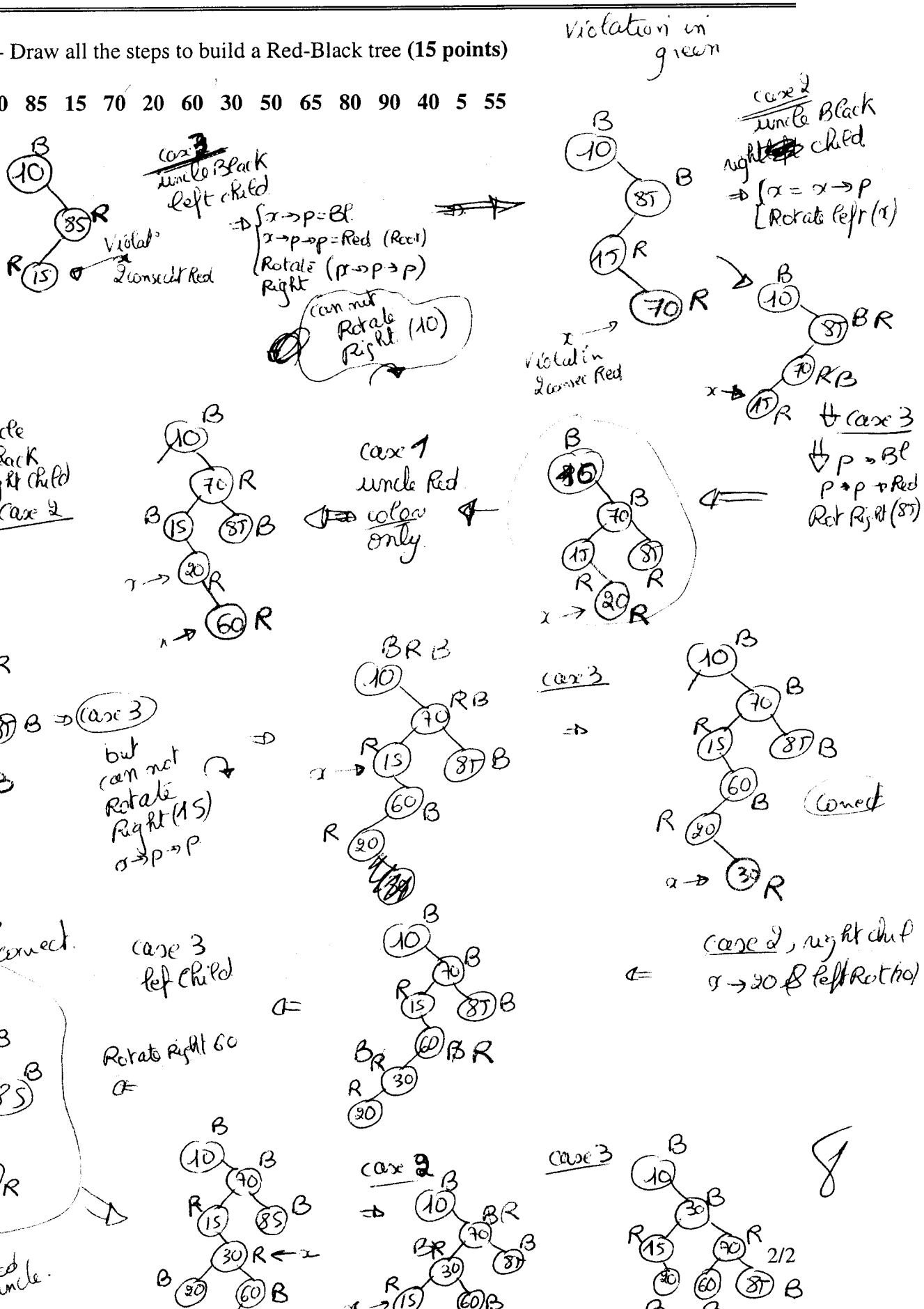
Correct.

15

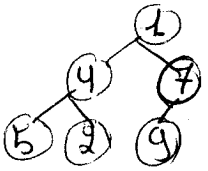
1/1

2- Draw all the steps to build a Red-Black tree **(15 points)**

10 85 15 70 20 60 30 50 65 80 90 40 5 55



3. Show the result of applying Build-Max-Heap to the array, 1; 4; 7; 5; 2; 9. Your answer should give the final contents of the **array**. (15 points)

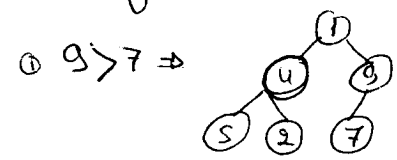


N=6 ⇒ start heapifying from Node 3 ⇒ (7)

steps in Heapify → ① compare both children, take Max Child

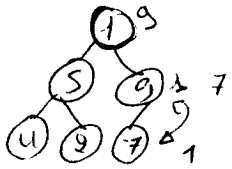
② if Max(child) > x ⇒ swap

③ Keep on till the end of (leaf)



① 9>7 ⇒

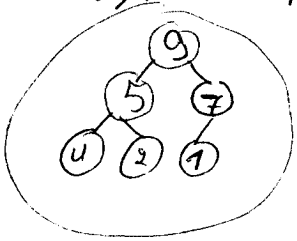
② 5>2 ⇒ MaxChild=5. 5>4 ⇒ swap



③ 9>5 ⇒ MaxChild=9

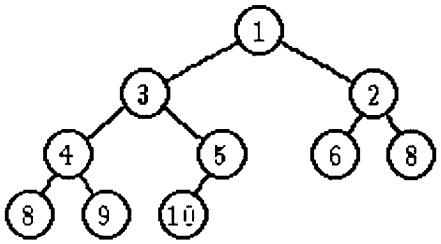
9>1 ⇒ swap 981 → at ① MaxChild is 7

& 1<7 ⇒ swap (1 8 7)



Max Heap

4.Consider the priority queue (heap) represented by the given height-biased leftist tree. Show the modified tree under each of the following operations. (Note: The two operations are independent. Each of them starts from the above tree.) **(15 points)**



Insertion of the key 7.

Deletion of a key.

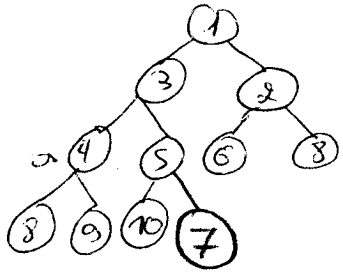
15

Insertion (7)

steps

① create a hole in the next available location

② percolate up all the way to the root if necessary



① 7 min child is 7
7 < 5 ⇒ Nothing

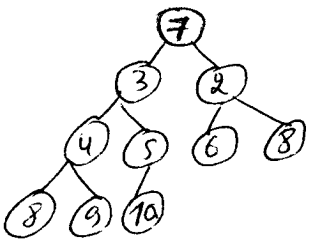
percolate up (Min Heap)

take Min child
if Min child < parent
⇒ swap
Repeat till Root.

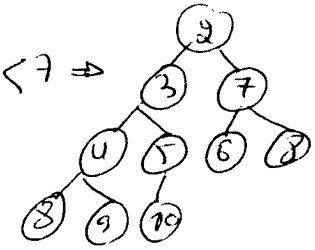
Deletion In heap deletion is done at the Root

Delete Root:
put last element in Root's place
percolate down all the way to a leaf

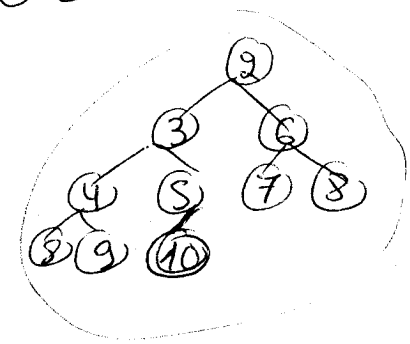
percolate Down
- find smallest child
- sm child < parent ⇒ swap



2 < 7 ⇒



6 < 7 ⇒

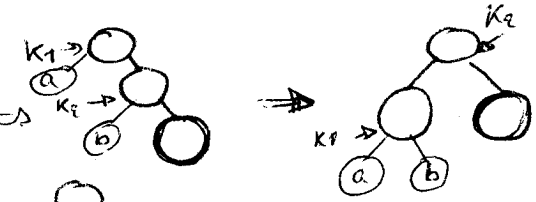


15

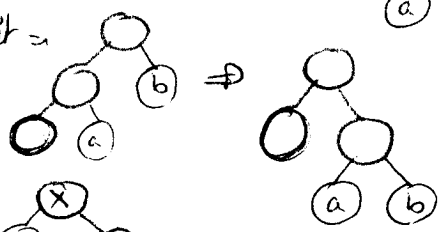
5. AVL: Explain single and double rotations used in AVL tree support your explanation by giving examples **(10 points)**

Single Rotation is done in 2 cases:

- adding right child to rightsubtree ⇒ Retate left ⇒

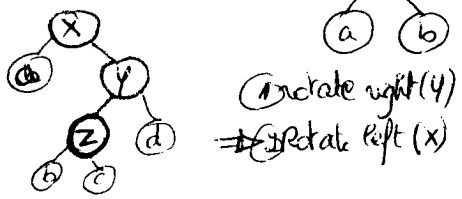


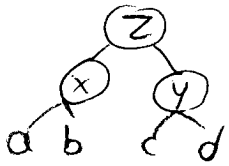
- adding left child to left subtree ⇒ Rotate right ⇒



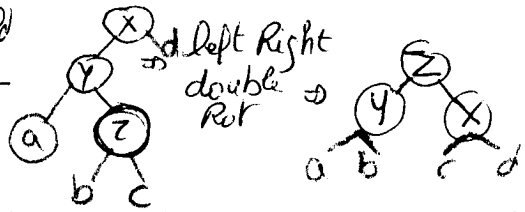
Double Rotation also 2 cases:

- adding left child to right subtree ⇒ right left Double Rot





- adding right child to left subtree



6. SHELL SORT using the sequences : 9 , 5 and 1 **(15 points)**

-9, 25, -3, 0, 14, 55, 12, 7, 4, -7, -8, 45, 3, 9, 5, 1

-9 25 -3 0 14 55 12 7 4 -7 -8 45 3 9 5 1

seq3) -9 -8 -3 0 9 5 1 7 4 -7 25 45 3 14 55 12

seq5) -9 -8 -3 0 9 5 1 7 4 -7 25 45 3 14 55 12

-9 -8 -3 0 -7 5 1 3 4 9 12 45 7 14 55 25

seq1) -9 -8 -3 -7 0 ⇒ [-9 -8 -7 -3] 0 5 1 3 4 9 12 45 7 14 55 25

-7 go left

-9 -8 -7 -3 0 1 5 3 4 9 - -

[-9 -8 -7 -3 0 1 3 4 5] 9 12 45 7 14 55 25

7<12 swap, 7<9 swap, 45>14 swap, 14>12 stop — go left ⇒ 5<7 stop

[" [7 9 12] 45 14 55 25

["] 7 9 12 14 45 55 25

-9 -8 -7 -3 0 1 3 4 5 7 9 12 14 45 55 25

55 < 25 swap

go left ⇒ 25 < 45 swap, 25 > 14 stop ⇒

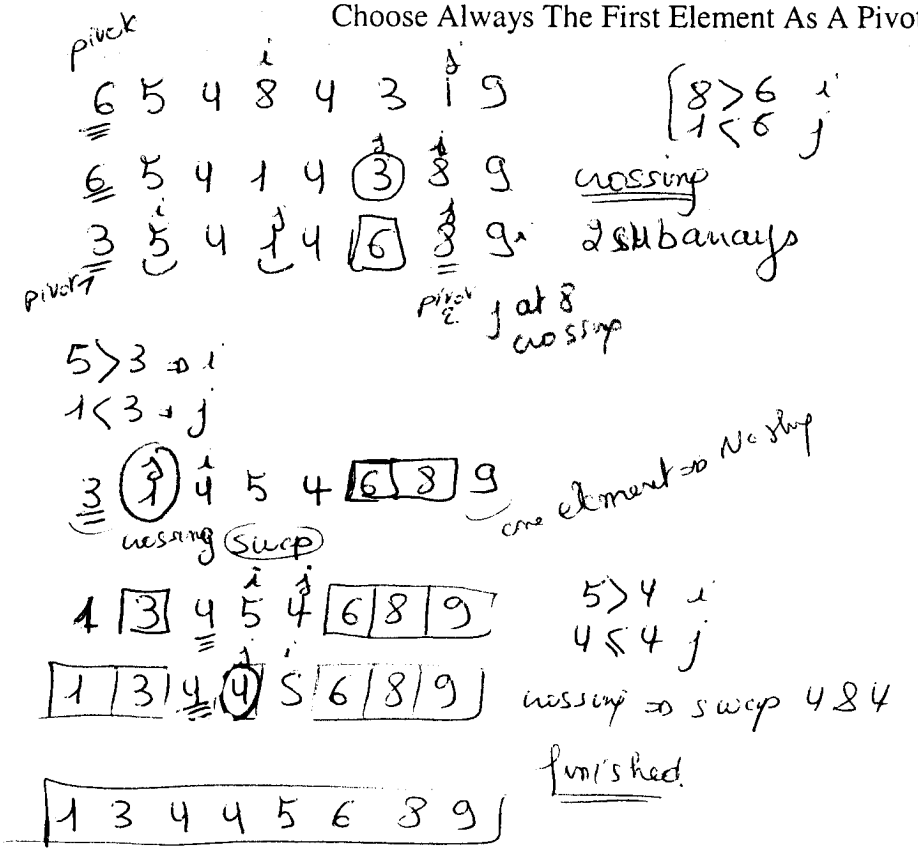
-9	-8	-7	-3	0	1	3	4	5	7	9	12	14	25	45	55

5/5

7. QUICK SORT **(15 points)**

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

Choose Always The First Element As A Pivot.



15

Good Luck