Naman Jain Vimal Kumar
1001670153

Problem 1 - Exercise 1.1

a. *What are the seven sorting strategies discussed in class? Identify them as having a run-time of θ(n), or θ(n log n) or θ (n^2)?*
b. *What are the number of inversion pairs in the following unsorted list? 6, 7, 3, 9, 2, 9, 3, 4, 2, 8, 8*
c. *Why is an in-place sorting algorithm more preferable to one that is not?*
d. *Apply Quick sort (using 3 medians) on this sequence: 7, 11, 14, 6, 9, 4, 3, 12 (show each step).*

Solution

a. Some of the sorting strategies are:

Algorithm	Average Case
Insertion Sort	O (n^2)
Merge Sort	O (n log(n))
Bubble Sort	O (n^2)
Heap Sort	O (n log(n))
Quick Sort	O (n log(n))
Bucket Sort	O (n + m)
Radix Sort	O (n log(m))

Notes:
Bubble Sort:

- Push the heaviest element to the bottom for every iteration
- Can keep tracks of swaps, if no swaps, the list is sorted
- Ref: https://www.youtube.com/watch?v=nmhjrI-aW5o

Insertion Sort:

- Sorts array by dividing the array into sorted and unsorted part
- Like playing of cards. 1st element of the unsorted list is compared to every element from sorted list to find its location.
- Ref: https://www.youtube.com/watch?v=OGzPmgsI-pQ

Merge Sort:

- Split and merge the array
- Merge sort is better than insertion sort when n > 30
- Ref: https://www.youtube.com/watch?v=JSceec-wEyw

Quick Sort:

- Partition list based on a pivot element where one lists consists of elements lesser and pivot and the other greater than the pivot, such that pivot location is found.
- Ref: https://www.youtube.com/watch?v=PgBzjlCcFvc

Heap Sort:

- Pushing of elements to a min heap and then popping elements one by one
- Ref: https://www.youtube.com/watch?v=MtQL_ll5KhQ

Naman Jain Vimal Kumar
1001670153

Bucket Sort:

- Create a HashMap. And assign true to all the matching index to that of array element. Read elements of hashmap, and we get the sorted list.
- Ref: https://www.youtube.com/watch?v=VuXbEb5ywrU

b. List: 6, 7, 3, 9, 2, 9, 3, 4, 2, 8, 8
Pairs:

(6, 7)	(6, 3)	(6, 9)	(6, 2)	(6, 9)	(6, 3)	(6, 4)	(6, 2)	(6, 8)	(6, 8)
	(7, 3)	(7, 9)	(7, 2)	(7, 9)	(7, 3)	(7, 4)	(7, 2)	(7, 8)	(7, 8)
		(3, 9)	(3, 2)	(3, 9)	(3, 3)	(3, 4)	(3, 2)	(3, 8)	(3, 8)
			(9, 2)	(9, 9)	(9, 3)	(9, 4)	(9, 2)	(9, 8)	(9, 8)
				(2, 9)	(2, 3)	(2, 4)	(2, 2)	(2, 8)	(2, 8)
					(9, 3)	(9, 4)	(9, 2)	(9, 8)	(9, 8)
						(3, 4)	(3, 2)	(3, 8)	(3, 8)
							(4, 2)	(4, 8)	(4, 8)
								(2, 8)	(2, 8)
									(8, 8)

Total number of pairs: ${}_{11}P_2 = 55$
From the above table, number of inversions (in red) = 25

c. In-place sort algorithm has a space complexity is 1, while in other algorithms it is n. This reduces the cost on hardware. Thus if 2 algorithms of same time complexity are chosen, then one with in-place is given priority.

d. List: 7, 11, 14, 6, 9, 4, 3, 12

Step 1)

Notes		Array							
	pivot	0	1	2	3	4	5	6	7
Select Pivot	9	7	11	14	6	9	4	3	12
min = 0 max = 6		7	11	14	6	12	4	3	
min++		7	11	14	6	12	4	3	
Swap 1, 6 min++ max--		7	3	14	6	12	4	11	
Swap 2, 5 min++ max--		7	3	14	6	12	4	11	
min++		7	3	4	6	12	14	11	
Swap pivot with 4		7	3	4	6	9	14	11	12

Step 2) Work on left side of the list (Insertion Sort as it is a small list)

Notes		Array							
		0	1	2	3	4	5	6	7
Swap 0, 1		7	3	4	6	9	14	11	12

Swap 1, 2		3	7	4	6	9	14	11	12
Swap 2, 3		3	4	7	6	9	14	11	12
		3	4	6	7	9	14	11	12

Step 3) Work on right side of the list (Insertion Sort as it is a small list)

Notes		Array							
		0	1	2	3	4	5	6	7
Swap 5, 6		3	4	6	7	9	14	11	12
Swap 6, 7		3	4	6	7	9	11	14	12
		3	4	6	7	9	11	12	14

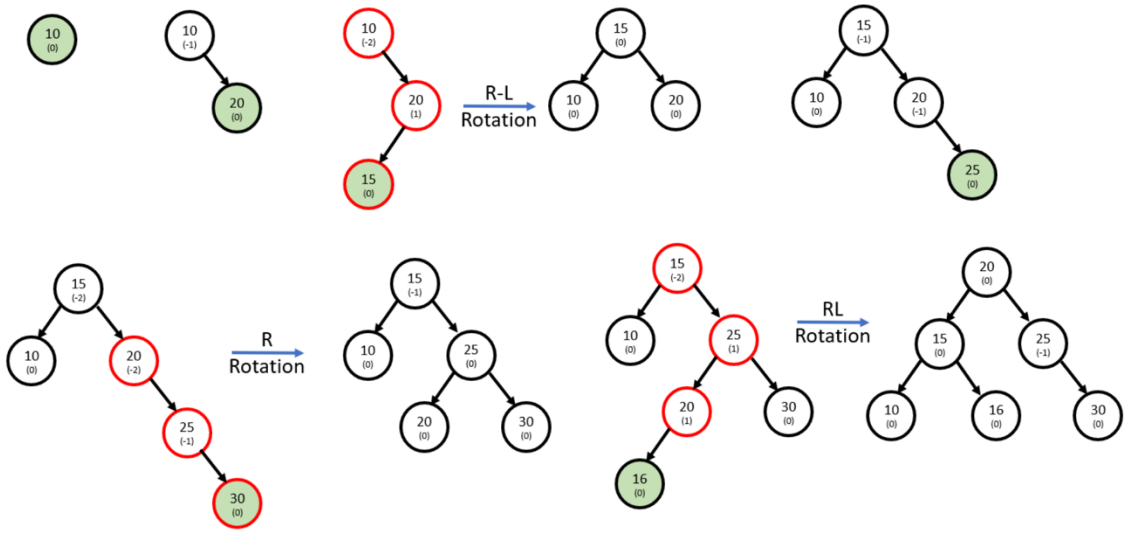
Sorted List: 3, 4, 6, 7, 9, 11, 12, 14

Problem 2

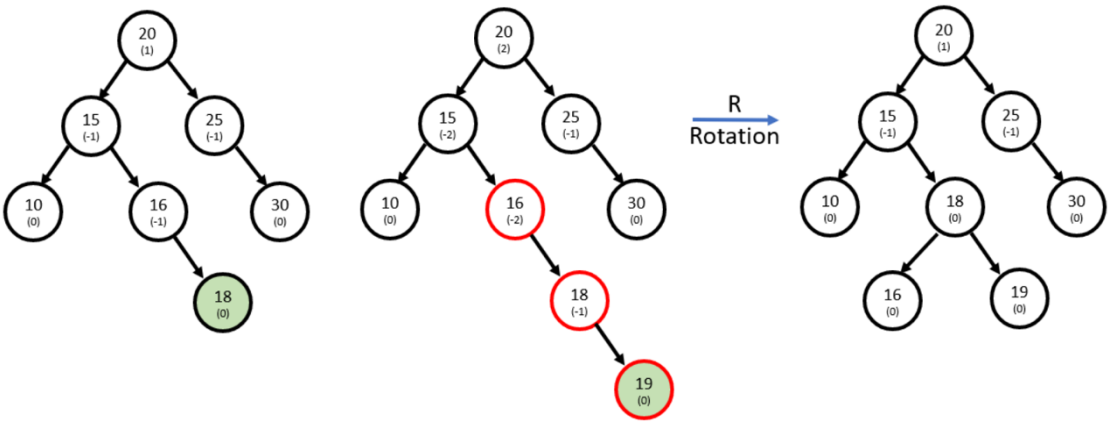
a. *Insert the following sequence of elements into an AVL tree, starting with an empty tree (show each step) : 10, 20, 15, 25, 30, 16, 18, 19.*
b. *Delete 30 in the AVL tree that you got(show each step).*
c. *What maximum difference in heights between the leaf s of a AVL tree is possible?*

Solution

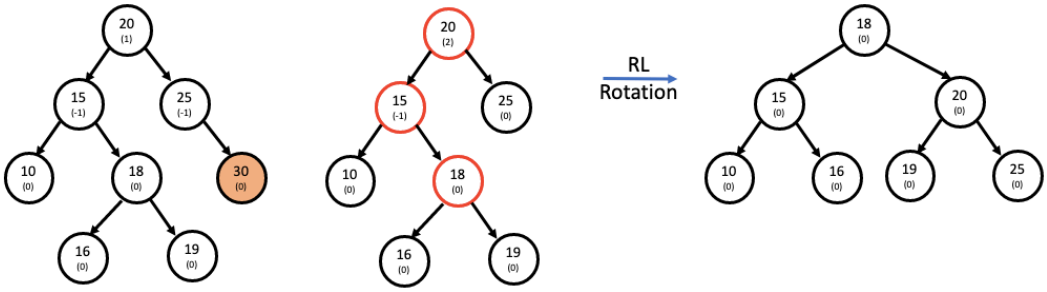
a.



Naman Jain Vimal Kumar
1001670153



b. Deleting 30 from the above tree:



c. The difference in height for a node in an AVL tree could be [-1, 0, +1]. Therefore, absolute max difference of heights in leaf's node is also equal to 1.

Problem 3

a. What are the operations that could be performed in O(log n) time complexity by red-black tree?

b. Insert the following sequence into a red black tree (show each step): 5, 6, 1, 9, 2, 4, 3, 8, 7 (20 p)

Solution

a. The operations that can be performed in O(log n) time complexity by red-black trees are:

ALGORITHM	AVERAGE	WORST CASE
SEARCH	O (log n)	O (log n)
INSERT	O (log n)	O (log n)
DELETE	O (log n)	O (log n)

b.

Naman Jain Vimal Kumar
1001670153