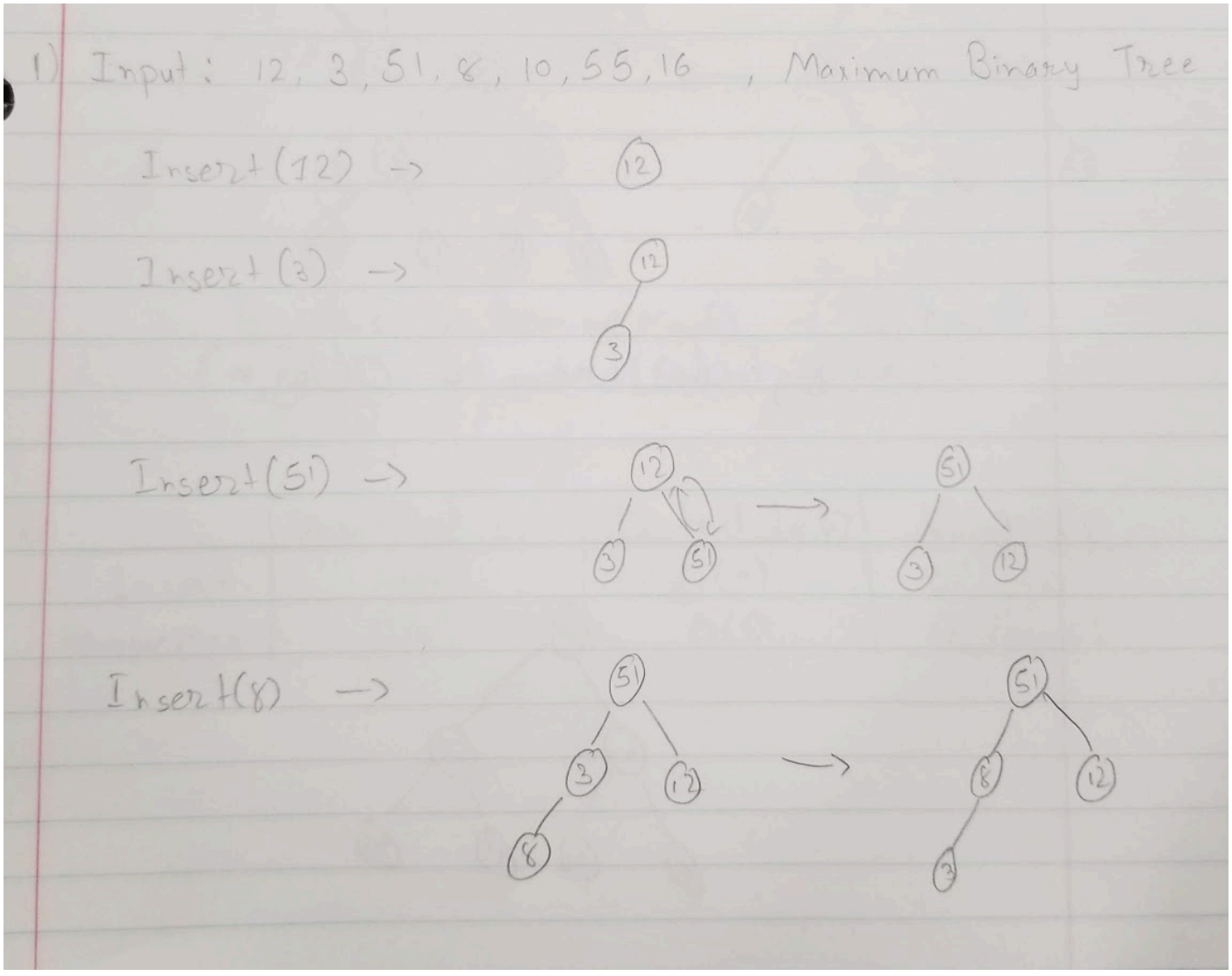
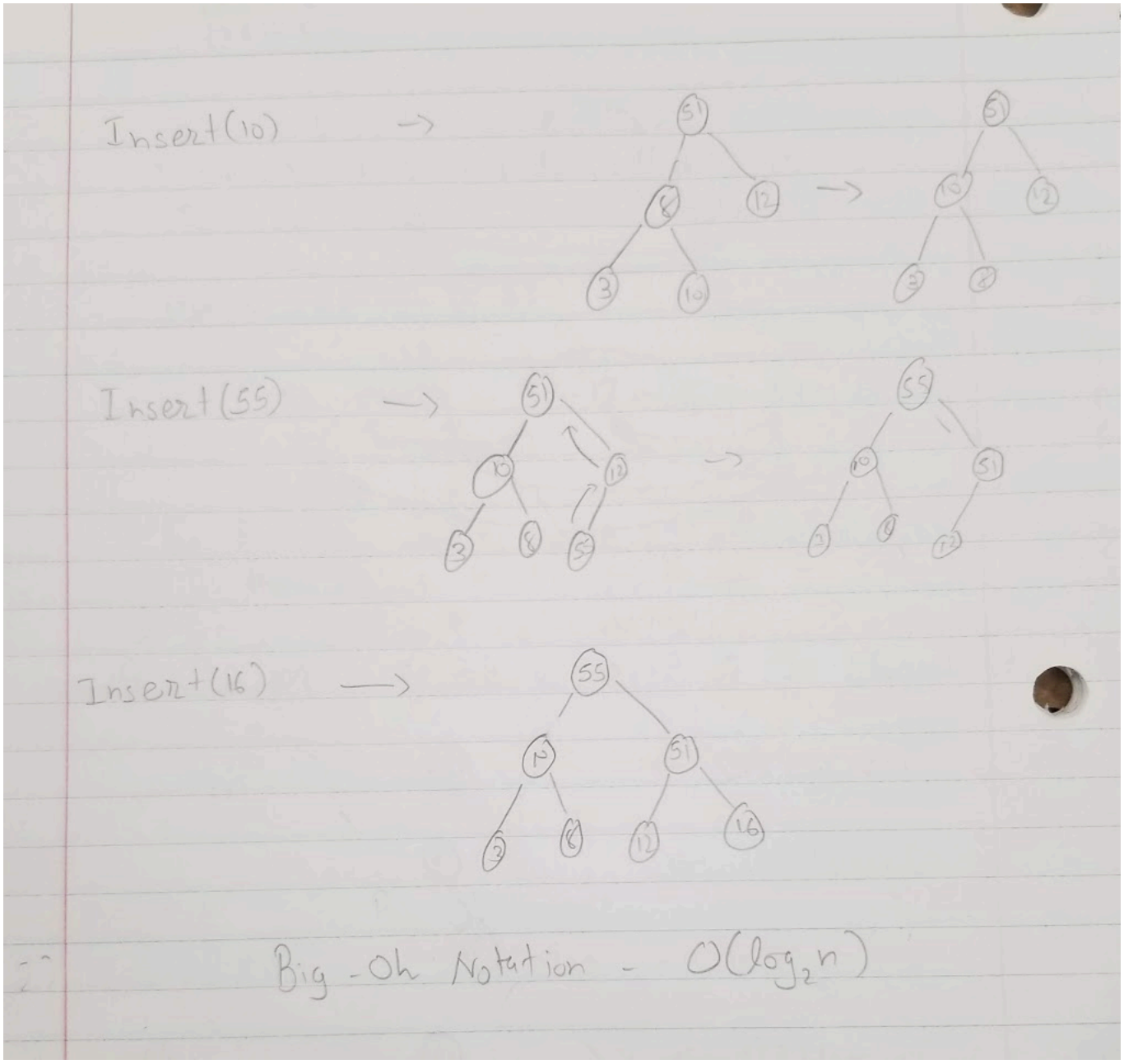
Pratik	Patel	501
First Name	Last Name	Section Number

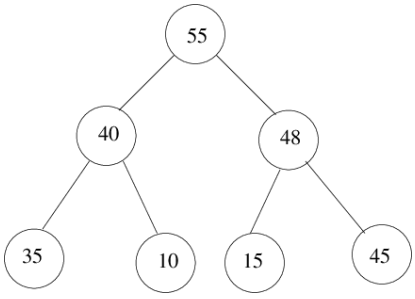
CPSC 221 Priority Queue and Heap

1. Represent the following input 12, 3, 51, 8, 10, 55, 16 in order to get a maximum binary heap. Illustrate each step of the algorithm. What is the Big-O running time of the algorithm?

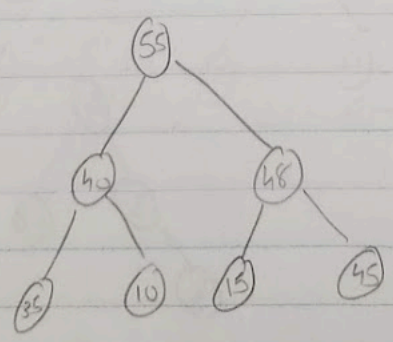




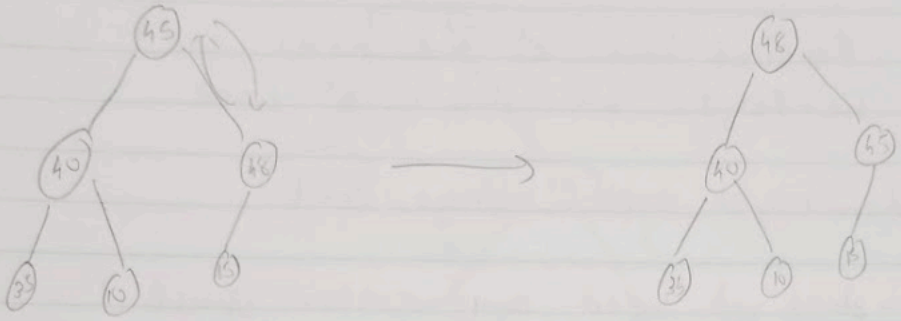
2. Illustrate a state of the heap below after the sequence of two operations: removeMax(), insert(50). What is the Big-O running time of these operations?



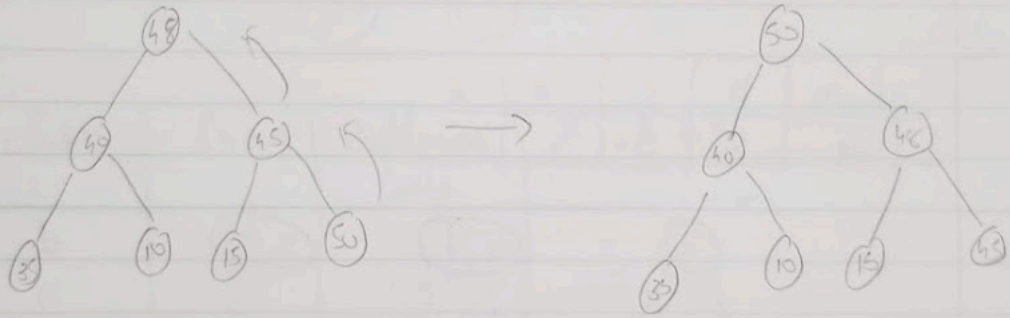
2) Given Tree



removeMax()



Insert (50)

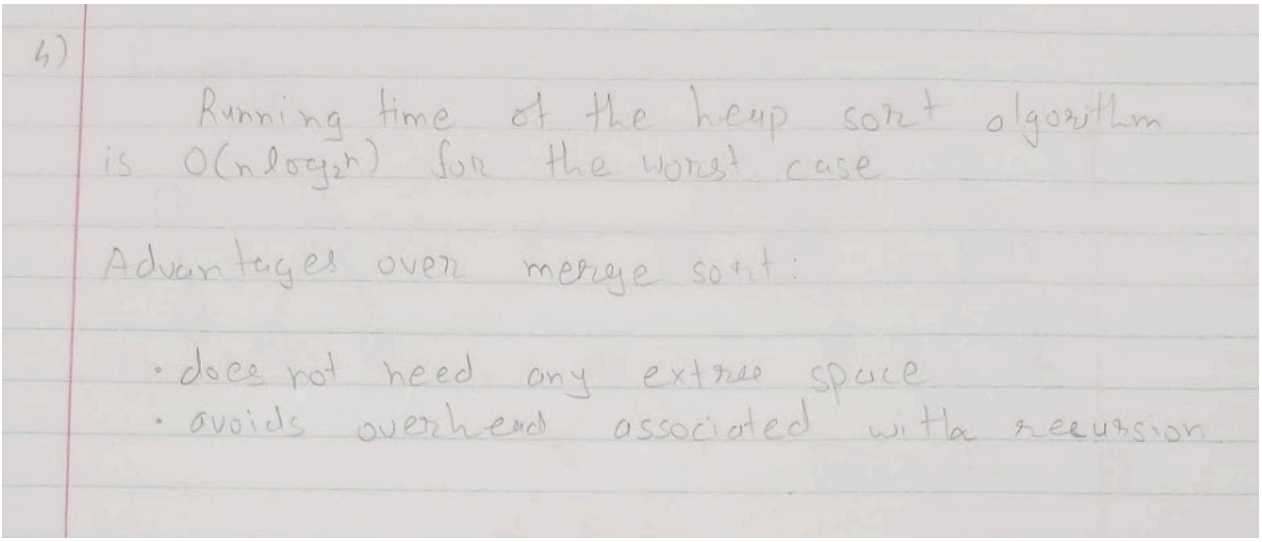


Big-Ohs i) removeMax() → $O(\log_2 n)$
ii) insert(50) → $O(\log_2 n)$

3. Consider three different implementations of Maximum Priority Queue based on an unsorted list, sorted list and heap. What is the running time (use Big-O notation) for the following operations in the table below?

	Unsorted List	Sorted List	Heap
Insert	O(1)	O(n)	O(logn)
Remove Max	O(n)	O(1)	O(logn)

4. What is the running time of the heap sort algorithm? What is the advantage of the heap sort over the merge sort algorithm?

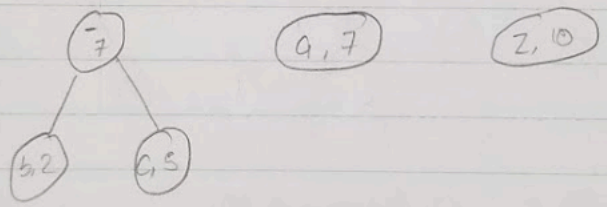


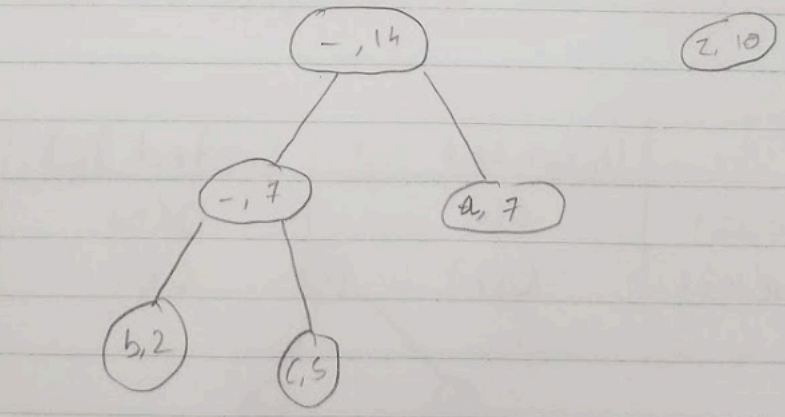
5. Find the Huffman codes for the characters in the table below. Assume that your minimum priority queue is represented as a sorted array.

character	b	c	a	z
frequency	2	5	7	10

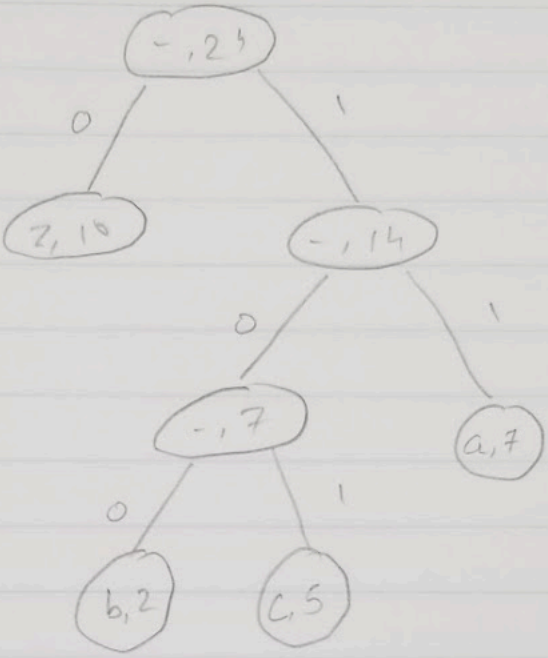
5)

char	b	c	a	z
freq	2	5	7	10





Final Tree



char	freq	Huffman Code	# of bits
b	2	100	6
c	5	101	15
a	7	11	14
z	10	0	10