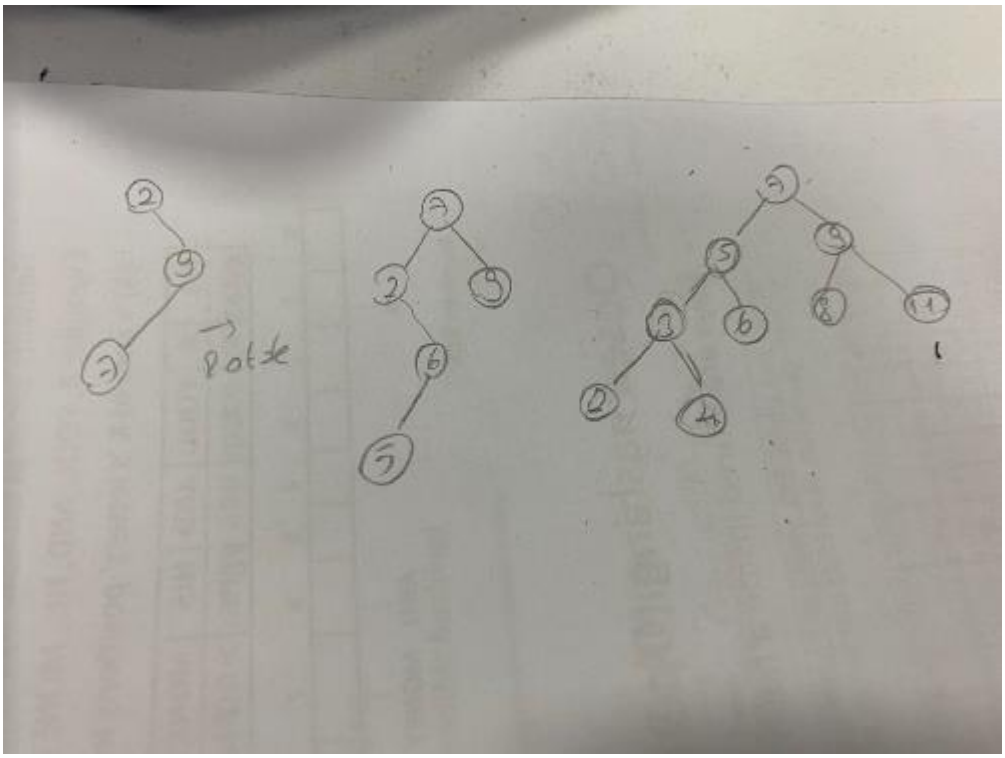
FSM	Mühendislik Fakültesi Bilgisayar Mühendisliği Programı **BLM307E Algorithm Analysis & Design Lab Work**	Adı Soyadı: Notu:

Lab 6: HeapSort – AVL Trees

Question 1: Draw a AVL tree for given sequence of numbers: 2, 9, 7, 6, 5, 8, 11, 4, 3



Question 2: Show the time efficiency classes of the priority queue's for following operations and data types.

	Unsorted Array	AVL Tree	Heap
find maximum element	O(1)	o(log n)	O(log n)
Delete maximum element	O(n)	o(log n)	O(log n)
Add new value	O(n)	o(log n)	O(1)

Question 3: Assume that we want to implement the priority-based job scheduler. Which is the best data structure to implement the priority-based operation system process scheduler (LinkedList, Queue, BST, etc.). What is the insertion performance of this data structure? Explain.

For example: I use an AVL Tree for process scheduler and, implement AVL Tree as an unsorted array. Insertion performance of the AVL Tree with an unsorted array is O(n).

I use Queue. Performance: O(1) . Queue Insertion operation is also called enqueue. Queue operations implement FIFO (First In First Out) principle. The element added at the beginning of the queue is deleted first. A new element can be added at the REAR of the queue.

Question 4: Why is the time complexity of Heap Sort O(n*Log n)? The time complexity of which operation is O(logn) and which one is O(n)?

we suggested that the basic heap operation of Heapify operates in O(log n) time because the heap has O(log n) levels, and the component being sifted progresses down one level of the tree after a fixed quantity of work. Therefore the total running time of HeapSort is O(n log n).

A common algorithm with O(log n) time complexity is Binary Search whose recursive relation is T(n/2) + O(1) i.e. at every subsequent level of the tree you divide the problem into half and do a constant amount of additional work.

O(1) indicates in fastened time - freelance of the number of things. O(N) denotes in proportion to the amount of things.

Question 5: Given a sequence of numbers: 2, 9, 7, 6, 5, 8

a) Draw a binary max-heap (in a tree form) by inserting the above numbers reading them from left to right
b) Show a tree that can be the result after the call to deleteMax() on the above heap