****

***INSTITUTE OF INFORMATION TECHNOLOGY***

***JAHANGIRNAGAR UNIVERSITY***

**Number of Assignment :** 01

**Course Tittle :** Data Structure

**Course Code :** ICT – 2101

**Submission Date :** 23/02/2021

**Submitted To Submitted By**

Dr. M. Abu Yousuf MD. Shakil Hossain

Professor Roll – 2023

IIT – JU 2nd year 1st Semester

IIT – JU

**Answer to the Question no – 01**

If we want time complexity for the problem as O(n log n), we can then simply apply merge sort and add the last k elements of the array.

**Pseudocode:**

* Declare left and right variables.
* Left will be assigned to 0 and right will be assigned to n-1.
* Find mid = (left + right / 2)
* Call merge sort on (left . mid) and (mid + 1 . rear)
* Then we will merge on the 2 sub problems.

[Sorting completed]

* Using Sum = 0, we will simply add the last ‘k’ element.

**Answer to the Question no - 02**

We can use an AVL tree.

* new(): create a new AVL tree, which takes O(1) time
* insert(x): use AVL insertion, which takes O(log n) time
* member(x): use BST membership testing, which takes O(log n) time
* increaseBy(x): use an inorder traversal to iterate through all nodes

of the tree. For each node, add x to the value. Note that this preserves

the relative order of all elements so will not break the AVL invariant. It

takes O(n) time because each node of the tree is visited once.

**THE END**