**Tutorial 8**

We will do Q 5 and Q 6 in mid-exam paper.

**Question 5.** Given an array of n ≥ 2 **distinct** integers (i.e., no two integers are the same) sorted in ascending order, say [x(1),...,x(n)], we want to find the absolute minimum *difference between the x(i) and i*. For example, for *x = [-10, 9, 10, 12, 13, 16] ,* the minimum *difference d =*|*x(2)−2| =* |*9−2| = 7*.

(a) Use a linear time algorithm to solve the problem.

(b) Use a divide and conquer approach to solve the problem. The running time should be O(logn).

(c) Set up and solve a recurrence equation for part (b) to estimate the running time of your algorithm. Prove that the running time of your algorithm is O(logn)

**Hint:** The difference will first decrease and then increase.

**Question 6.**

Suppose we have an array of n positive integers. A contiguous subarray A[i .. j] is called a squared interval if the sum of its entries is a squared number. Design a greedy algorithm to compute the maximum number of squared intervals such that every entry in A will be covered at most once.

1. You can state your algorithm in English or in Pseudo code
2. What is the running time of algorithm in big-O.
3. Prove that your algorithm is correct.

4, squared

16, squared

36, squared

25, squared

1 7 7 5 5 3 4 7 6 9 1 13 5