COL 351 : Analysis and Design of Algorithms

Tutorial Sheet - 6

Question 1 Device a divide-and-conquer algorithm to merge k sorted arrays, each with n elements, into a single sorted array of kn elements. What is the time complexity of this algorithm, in terms of k and n?

Question 2 You are given an n-node complete binary tree T of height h, so $n = 2^h - 1$. The nodes of T are labelled with distinct real numbers. A node in T is a local minimum if its label is smaller than the label of its neighbours. Device an algorithm to find a local minimum of T in $O(\log n)$ time.

Question 3 Given an n sized array A, the *Inversion Count* of A is the number of pairs (i, j) such that $A[i] > A[j]$ and $i < j$. So if A is already sorted, then the inversion count is 0, but if A is sorted in the reverse order, the inversion count is ${}^nC_2$. Device a divide-and-conquer algorithm to compute *Inversion Count* of an array A of size n in $O(n \log n)$ time.

Hint: Use ideas from Merge Sort.

Question 4 Suppose you have to multiply two n bit positive integers x and y. Observe that x can be written as $x_1 \cdot 2^{n/2} + x_0$, for appropriate x_0, x_1. Similarly, y can be written as $y_1 \cdot 2^{n/2} + y_0$, for appropriate y_0, y_1. Then,

$$\begin{aligned} xy &= x_1 y_1 \cdot 2^n + (x_1 y_0 + x_0 y_1) \cdot 2^{n/2} + x_0 y_0 \\ &= x_1 y_1 \cdot 2^n + (p - x_1 y_1 - x_0 y_0) \cdot 2^{n/2} + x_0 y_0 \end{aligned}$$

where, $p = (x_1 + x_0)(y_1 + y_0)$. Use this observation to device an $O(n^{\log_2 3}) = O(n^{1.59})$ time algorithm to multiply two n bit positive integers.

Question 5 Let A be an array of size n with integer entries. Device an $O(n)$ time algorithm to check if there exists an $x \in A$ such that count of x in A is at least $\lceil n/10 \rceil$.