CSCI 570 - Fall 2016 - HW 5

September 23, 2016

1. Solve the following recurrences by giving tight Θ-notation bounds in terms of n for sufficiently large n. Assume that T represents the running time of an algorithm, i.e. $T(n)$ is positive and non-decreasing function of n and for small constants c independent of $n, T(c)$ is also a constant independent of n.

   (a) $T(n) = 4T(n/2) + n^2 \log n$

   (b) $T(n) = 8T(n/6) + n \log n$

   (c) $T(n) = 2T(\sqrt{n}) + \log n$

2. Solve Kleinberg and Tardos, Chapter 5, Exercise 3.

3. Solve Kleinberg and Tardos, Chapter 5, Exercise 5.

4. Assume that you have a black-box that can multiply two integers. Describe an algorithm that when given an n-bit positive integer a and an integer x, computes x^a with at most $\mathcal{O}(n)$ calls to the black-box.

5. Consider the following algorithm *StrangeSort* which sorts n distinct items in a list A.

   (a) If $n \leq 1$, return A unchanged.

   (b) For each item $x \in A$, scan A and count how many other items in A are less than x.

   (c) Put the items with count less than $n/2$ in a list B.

   (d) Put the other items in a list C.

   (e) Recursively sort lists B and C using *StrangeSort*.

   (f) Append the sorted list C to the sorted list B and return the result.

   Formulate a recurrence relation for the running time $T(n)$ of *StrangeSort* on a input list of size n. Solve this recurrence to get the best possible $O(.)$ bound on $T(n)$.

6. Consider an array A of n numbers with the assurance that $n > 2, A[1] \geq A[2]$ and $A[n] \geq A[n-1]$. An index i is said to be a local minimum of the array A if it satisfies $1 < i < n, A[i-1] \geq A[i]$ and $A[i+1] \geq A[i]$.

(a) Prove that there always exists a local minimum for A.

(b) Design an algorithm to compute a local minimum of A. Your algorithm is allowed to make at most $O(\log n)$ pairwise comparisons between elements of A.

7. Given a sorted array of n integers that has been rotated an unknown number of times, give an $O(\log n)$ algorithm that finds an element in the array. An example of array rotation is as follows: original sorted array $A = [1,\ 3,\ 5,\ 7,\ 11]$, after first rotation $A' = [3,\ 5,\ 7,\ 11,\ 1]$, after second rotation $A'' = [5, 7,\ 11,\ 1,\ 3\ ]$.