CSCI570 - Spring 2018 - HW5

Due Feb. 14, 11:59pm

This homework assignment covers divide and conquer algorithms and recurrence relations. It is recommended that you read all of chapter 5 from Klienberg and Tardos, the Master Theorem from the lecture notes, and be familiar with the asymptotic notation from chapter 2 in Klienberg and Tardos.

1 Graded Problems

1. The recurrence $T(n) = 7T(n/2) + n^2$ describes the running time of an algorithm ALG. A competing algorithm $ALG^{'}$ has a running time of $T^{'}(n) = aT^{'}(n/4) + n^2 \log n$. What is the largest value of a such that $ALG^{'}$ is asymptotically faster than ALG?

2. Solve the following recurrences by giving tight Θ-notation bounds in terms of n for sufficiently large n. Assume that $T(\cdot)$ 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. Note that some of these recurrences might be a little challenging to think about at first.

   (a) $T(n) = 4T(n/2) + n^2 \log n$
   (b) $T(n) = 8T(n/6) + n \log n$
   (c) $T(n) = \sqrt{6006}T(n/2) + n^{\sqrt{6006}}$
   (d) $T(n) = 10T(n/2) + 2^n$
   (e) $T(n) = 2T(\sqrt{n}) + log_2 n$
   (f) $T^2(n) = T(n/2)T(2n) - T(n)T(n/2)$
   (g) $T(n) = 2T(n/2) - \sqrt{n}$

3. 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 $O(\cdot)$ bound on $T(n)$.

4. Solve Kleinberg and Tardos, Chapter 5, Exercise 1.

2 Practice Problems

1. Solve Kleinberg and Tardos, Chapter 5, Exercise 3.
2. Solve Kleinberg and Tardos, Chapter 5, Exercise 6.
3. 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.
4. A polygon is called convex if all of its internal angles are less than 180° and none of the edges cross each other. We represent a convex polygon as an array V with n elements, where each element represents a vertex of the polygon in the form of a coordinate pair (x, y). We are told that $V[1]$ is the vertex with the least x coordinate and that the vertices $V[1], V[2], \cdots, V[n]$ are ordered counter-clockwise. Assuming that the x coordinates (and the y coordinates) of the vertices are all distinct, do the following.
   (a) Give a divide and conquer algorithm to find the vertex with the largest x coordinate in $O(\log n)$ time.
   (b) Give a divide and conquer algorithm to find the vertex with the largest y coordinate in $O(\log n)$ time.
5. 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 $A^{''} = [5, 7, 11, 1, 3]$.