Homework 6

The purpose of this homework is to give you practice with Divide and Conquer, to prepare you for Exam 2. **There is no quiz associated with this homework.**

Note that Exam 2 is cumulative, while this homework covers only Divide and Conquer.

Master Theorem Reminder:

Let $T(n)$ be defined by the recurrence $T(n) = aT(n/b)+f(n)$ (where $a \geq 1$ and $b > 1$ are constants). Then $T(n)$ can be bounded asymptotically as follows:

1. If $f(n) = O(n^{\log_b a-\epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$
3. If $f(n) = \Omega(n^{\log_b a+\epsilon})$ for some constant $\epsilon > 0$ and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$.

Problem 1: Master Method

Use the Master Theorem above to give a tight asymptotic bound for each of the following recurrences, or argue why it doesn't apply.

1. $T(n) = 3T(n/2) + \Theta(n)$
2. $T(n) = 3T(n/2) + \Theta(n^2)$
3. $T(n) = 16T(n/2) + \Theta(n^3 \lg n)$
4. $T(n) = 8T(n/2) + \Theta(n^3 \lg n)$

Problem 2: Value matches position

Suppose you are given a sorted sequence of *distinct* integers $\{a_1, a_2, \ldots a_n\}$. Give an $O(\log n)$ algorithm to determine whether there exists an index i such that $a_i = i$. For example, in $\{-10, -3, 3, 5, 7\}$, $a_3 = 3$; there is no such i in $\{2, 3, 4, 5, 6, 7\}$. Write the recurrence for your algorithm and show that its recurrence solves to $O(\log n)$ (e.g., using the Master Method).

Problem 3: Circular shift

Suppose you are given an array A of n sorted numbers that has been *circularly shifted* to the right by k positions. For example $\{35, 42, 5, 15, 27, 29\}$ is a sorted array that has been circularly shifted $k = 2$ positions, while $\{27, 29, 35, 42, 4, 15\}$ has been shifted $k = 4$ positions. Give an $O(\log n)$ algorithm to find the largest number in A. You may assume the elements of A are distinct. Write the recurrence for your algorithm and show that its recurrence solves to $O(\log n)$ (e.g., using the Master Method).

Problem 4: Maximum sum

Given a list of integers $a_1, a_2, \ldots, a_n$ we are interested in finding a subsequence having maximum sum; i.e., if for $i \leq j$ we define $A_{i,j} = \sum_{i \leq k \leq j} a_k$, we want i, j such that $A_{i,j}$ is maximum. For example, if the given list is $\{1, -5, 1, 9, -7, 9, -4\}$, the maximum subsequence is $A_{3,6}$. Give a divide-and-conquer algorithm for this problem. Write the recurrence for your algorithm and solve for its asymptotic upper bound (e.g., using the Master Method).