MCS 401: Computer Algorithms I (Fall 2020)
Homework 4

Due at 2:00pm CST, Monday, Nov 2

1. Give an asymptotic tight bound for $T(n)$ defined by each of the following recurrences. You can assume $T(1) = 1$ and n is a power of 2. (The Master Theorem is provided on the last page.)

   (a) $T(n) = 2\,T(n/2) + n^{1.5}$.

   (b) $T(n) = 3\,T(n/2) + n$.

   (c) $T(n) = 4\,T(n/2) + n^2$.

2. Suppose $T(1) = 1$. Use mathematical induction to upper bound the following recurrences.

   (a) If $T(n) \leq 2\,T(n-1) + 1$, then $T(n) = O(2^n)$.

   (b) $T(n) \leq T(\lfloor \frac{7}{10}n \rfloor) + T(\lfloor \frac{1}{5}n \rfloor) + n$, then $T(n) = O(n)$.
   ($\lfloor x \rfloor$ is the largest integer less than or equal to x.)

3. (2 points) Consider the *selection* problem: given an array of n numbers $(a_1, a_2, \ldots, a_n)$ and an integer $1 \leq k \leq n$, find the k-th smallest number in this array.

   A simple algorithm is to first sort the array in $O(n \log n)$ time and then output the k-th element. In this exercise, we will try to derive an $O(n)$ algorithm for this problem. Without loss of generality, we assume $(a_1, a_2, \ldots, a_n)$ are distinct and n is a power of 2.

   To make things easier, you are given access to an algorithm $\mathcal{A}$ which can find the median in linear time. More specifically, for any even number $t > 0$, you can call algorithm $\mathcal{A}$ on an array of t distinct numbers $(b_1, b_2, \ldots, b_t)$. Algorithm $\mathcal{A}$ will run in time $O(t)$ and output a number x such that $|\{i : b_i \leq x\}| = t/2$.

   (a) Design a divide-and-conquer algorithm for the selection problem. Your algorithm should run in $O(n)$ time.

   (b) Prove the correctness of your algorithm and analyze its running time.

The Master Theorem. Fix an integer $a \geq 1$ and real numbers $b > 1$, $c \geq 0$, $d > 0$.

Consider the following recurrence:

$$T(n) = \begin{cases} a \cdot T(n/b) + n^c & \text{if } n > 1 \text{ ,} \\ d & \text{if } n = 1 \text{ .} \end{cases}$$

For any n that is a power of b, we have

$$T(n) = \begin{cases} \Theta(n^c) & \text{if } \log_b a < c \text{ ,} \\ \Theta(n^c \log n) & \text{if } \log_b a = c \text{ ,} \\ \Theta(n^{\log_b a}) & \text{if } \log_b a > c \text{ .} \end{cases}$$