# CS 173, Spring 2009
# Homework 7

Due in class on Friday, April 3rd, 2009
(Total point value: 30 points.)

1. **Induction with inequalities [10 points]**

   Use induction to show that the following holds for all integers $n \geq 8$.

   $$n^2 > 7n + 1$$

2. **Unrolling [10 points]**

   As we have seen (or will soon see) in lecture, it takes $\theta(n \log n)$ time to sort a list of numbers. The usual way to find the median number in a list is to sort the list and then extract the element in the middle of the sorted list. However, there is a somewhat complex alternative algorithm whose running time is (simplifying slightly) given by the recurrence

   $$T(1) = 1$$

   $$T(n) = T(\frac{7}{10}n) + n$$

   (a) Unroll this recurrence (showing your work) and express $T(n)$ as a summation.

   (b) In lecture 22, Eric mentioned a couple types of summations whose closed forms you should know. See also the table on p. 157 of the textbook. Find an appropriate formula and convert your answer from part (a) into a closed form.

   (c) Express $T(n)$ in big-O notation, i.e. using $O$, $\Omega$ and/or $\theta$. (For example, is $T(n) = \theta(n)$? is it $\theta(n \log n)$? is it $\theta(n^2)$?)

   (d) How does the big-O running time of this alternate algorithm compare to the $\theta(n \log n)$ time of finding the median by sorting. (Is it larger? smaller? the same?)

3. **Algorithm analysis (10 points)**

The following algorithm takes as input an arbitrary list of $n$ real numbers $a_1$,...,$a_n$ and an integer $k$. The output is returned in the variable $selected$. The command **swap**$(a_i, a_j)$ is used to interchange the positions of $a_i$ and $a_j$ in the list. You can assume that **swap**$(a_i, a_j)$ takes $\Theta(1)$ time.

```
procedure select( a_1,..., a_n, k)
for i := 1 to k
begin
    min := i
    for j := i + 1 to n
        if a_j < a_min then
        begin
            min := j
            swap(a_i, a_min)
        end
end
selected := a_k
```

(a) If the initial list is $10, 5, 2, 8, 3, 0, 15$ with $k = 4$ what value does $selected$ have at the end?

(b) Suppose each line of the pseudo-code above is a $\Theta(1)$ operation. Suppose also that we are considering "worst-case" running time, so that the conditional code in the 6th line always executes. How many operations are executed by the algorithm for input consisting of a list of $n$ numbers and a specific value $k$? Express your answer as a function of $n$ and $k$ using big-theta notation and show how you derived that function.