CSE 102
Homework Assignment 4

1. Assume the correctness of the algorithm $\mathrm{Partition}(A, p, r)$ on page 171 of the text. Use induction to prove the correctness of $\mathrm{Quicksort}(A, p, r)$ on page 171.

2. Recall the n^{th} harmonic number was defined to be $H_n = \sum_{k=1}^{n} \left(\frac{1}{k}\right) = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n-1} + \frac{1}{n}$. Use induction to prove that

$$\sum_{k=1}^{n} kH_k = \frac{1}{2}n(n+1)H_n - \frac{1}{4}n(n-1)$$

for all $n \geq 1$. (Hint: Use the fact that $H_n = H_{n-1} + \frac{1}{n}$.)

3. Use the results of problem #2 above, and problem #3 on the Midterm 1 to show, by direct substitution, that the solution to the recurrence

$$t(n) = (n-1) + \frac{2}{n} \cdot \sum_{k=1}^{n-1} t(k)$$

is given by: $t(n) = 2(n+1)H_n - 4n$.

4. Design a recursive algorithm called $\mathrm{Extrema}(A, p, r)$ that, given an array $A[1 \cdots n]$ finds and returns both the min and max of the subarray $A[p \cdots r]$ as an ordered pair: $(\min(A[p \cdots r]), \max(A[p \cdots r]))$. Your algorithm should perform exactly $\lceil 3n/2 \rceil - 2$ array comparisons on an input array of length n. (Hint: Section 9.1 of the text describes an iterative algorithm that does this.)

   a. Write your algorithm in pseudo-code.
   b. Prove the correctness of your algorithm by induction on $m = r - p + 1$, the length of the subarray $A[p \cdots r]$.
   c. Write a recurrence for the number of comparisons performed on $A[1 \cdots n]$, and show that $T(n) = \lceil 3n/2 \rceil - 2$ is its solution.

5. (This is a slight re-wording of problem 8-5 on page 207 of CLRS 3rd edition.)
   Suppose that, instead of sorting an array, we just require that the elements increase on average. More precisely, we call an n-element array $A[1 \cdots n]$ ***k-sorted*** if for all i in the range $1 \leq i \leq n - k$, the following holds:

$$\frac{\sum_{j=i}^{i+k-1} A[j]}{k} \leq \frac{\sum_{j=i+1}^{i+k} A[j]}{k}$$

   a. What does it mean for an array to be 1-sorted?
   b. Give a permutation of the numbers $\{1, 2, 3, \dots, 10\}$ that is 2-sorted, but not sorted.
   c. Prove that an n-element array is k-sorted if and only if $A[i] \leq A[i+k]$ for all i in the range $1 \leq i \leq n - k$.
   d. Describe an algorithm that k-sorts an n-element array in time $\Theta(n \log n)$.