These are practice problems for the upcoming midterm exam. You will be given a sheet of notes for the exam. Also, go over your homework assignments. **Warning:** This does not necessarily reflect the length, difficulty, or coverage of the actual exam.

Problem 1. Solve the following recurrences exactly using the tree method. You may assume n is "nice". Prove your answers using mathematical induction.

(i) $T(n) = 2T(n/2) + n^3$, $T(1) = 1$.

(ii) $T(n) = T(\sqrt{n}) + 1$, $T(2) = 1$.

(iii) $T(n) = 2T(n/2) + n \lg n$, $T(1) = 1$.

(iv) $T(n) = T(n-3) + 5$, $T(1) = 2$.

Problem 2. Show that

(a)

$$\frac{1}{2} \le \sum_{j=1}^{\infty} \frac{1}{j2^j} \le 1$$

(b)

$$1 \le \sum_{j=1}^{\infty} \frac{1}{j^2} \le 2$$

Problem 3. Assume that you run quicksort, but always partition on the $n/3$rd smallest element. Analyze how many comparisons the algorithm does. Just get the exact value for the high order term. You do not have to worry about floors and ceilings and you can make reasonable simplifying assumptions.

Problem 4. Assume you have a list of n elements where every number is within k positions of its correct location, for some constant k.

(a) Give an algorithm that sorts this list with as few comparisons as possible (as a function of n and k). Just get the high order term right. How many comparisons does your algorithm use?

(b) Show that your algorithm is optimal using a decision tree argument.

Problem 5. Let $A[1, .., n]$ be an array of n numbers (some positive and some negative).

(a) Give an algorithm to find which *three* numbers have sum closest to zero. Make your algorithm as efficient as possible. Write it in pseudo-code.

(b) Analyze its running time.