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. Assume the you have a heap of size $n = 2^m - 1$ (stored in an array). (The largest element is at the root.) You may describe the following algorithms in high level language, but they *must* be clear. Give your answers as a function of n. You may assume n is large. You may use extra memory.

(a) Give an algorithm, minimizing the number of comparisons, to find the smallest element in the heap. Exactly how many comparisons does it take.

(b) Give an algorithm, minimizing the number of comparisons, to find the second smallest element in the heap. Exactly how many comparisons does it take.

(c) Give an algorithm, minimizing the number of comparisons, to find the third smallest element in the heap. Exactly how many comparisons does it take.

Problem 2. 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 3. Show that

(a)

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

(b)

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

Problem 4. 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.

Problem 5. Consider the following often suggested code for randomly permuting an array:

```
for i = 1 to n do
    j ← random(1,n)
    A[i] ↔ A[j]
end for
```

Prove that it does not produce a uniformly distributed random permutation.