Problem Set 2
(due Monday, October 12)

Problem 1 Suppose you have an array $A[1..n]$ of items. Each item has a color field (among other fields), which can equal red, white, or blue. Design an efficient algorithm to rearrange the array so that all the red items come first, followed by all the white items, followed by all the blue items. Your algorithm should make only one pass through the array, and should run in $O(n)$ time. Your algorithm should use only a constant amount of extra space (for example, you may not make a copy of the array).

Present your algorithm either in pseudocode or in actual code.

Problem 2 A word is an anagram of another word if we can permute the letters of the second word to obtain the first. For example, the words “deposit”, “dopiest”, “posited”, and “topside” form an anagram class; so do the words “microphotographic” and “photomicrographic”. Imagine you are given an online English dictionary with about 70,000 words, one word per line. Design an algorithm to output all the anagram classes, one class per line. Solve the problem as efficiently as you can.

Hint: Try sorting.

Problem 3 Design a comparison-based algorithm to find the second-smallest of $n$ items. Your algorithm should use at most $n + \lceil \log n \rceil - 2$ comparisons.

Hint: First find the smallest item by divide-and-conquer. How many candidate items are left for the second-smallest position?

Problem 4 Suppose you have an array of $n$ items. You are trying to determine whether a majority (more than $\frac{n}{2}$) of the items have the same value. In other words, does there exist a value that appears more than $\frac{n}{2}$ times? Describe an efficient algorithm to solve this problem. How much time does your algorithm take?