Problem Set 9

Reading: Chapters 32.1–32.2, 30.1–30.2, 34.1–34.2, 35.1

Both exercises and problems should be solved, but *only the problems* should be turned in. Exercises are intended to help you master the course material. Even though you should not turn in the exercise solutions, you are responsible for material covered in the exercises.

Mark the top of each sheet with your name, the course number, the problem number, your recitation section, the date and the names of any students with whom you collaborated.

Three-hole punch your paper on submissions.

You will often be called upon to "give an algorithm" to solve a certain problem. Your write-up should take the form of a short essay. A topic paragraph should summarize the problem you are solving and what your results are. The body of the essay should provide the following:

1. A description of the algorithm in English and, if helpful, pseudocode.
2. At least one worked example or diagram to show more precisely how your algorithm works.
3. A proof (or indication) of the correctness of the algorithm.
4. An analysis of the running time of the algorithm.

Remember, your goal is to communicate. Full credit will be given only to correct algorithms that are *which are described clearly*. Convoluted and obtuse descriptions will receive low marks.

Exercise 9-1. On-line String Matching

Recall that in an on-line algorithm, the input is generated as the algorithm is running. The idea is to solve the problem efficiently before seeing all the input. You can't scan forward to look at 'future' input, but you can store all input seen so far, or some computation on it.

(a) In this setting, the text $T[1 \dots n]$ is being broadcast on the network, one letter at a time, in the order $T[1], T[2], \dots$. You are interested in checking if the text seen so far contains a pattern P, where P has length m. Every time you see the next letter of the text T, you want to check if the text seen so far contains P.

Design an algorithm that solves this problem efficiently. Your algorithm should use no more than $\Theta(m)$ time on preprocessing P. In addition it should do only constant amount of work per letter received. Your algorithm can be randomized, with constant probability of correctness.

(b) Now say that you have the same pattern P, but the text $T[1 \dots n]$ is being broadcast in reverse. That is, in the order $T[n], T[n-1], \dots$ Modify your algorithm so that it still detects the occurence of P in the text $T[i \dots n]$ immediately (i.e., in constant time) after the letter $T[i]$ is seen.

Exercise 9-2. Some Summations

Assume you are given two sets $A, B \subset \{0 \ldots m\}$. Your goal is to compute the set $C = \{x + y : x \in A, y \in B\}$. Note that the set of values in C could be in the range $0 \ldots 2m$.

Your solution should run in time $O(m \log m)$ (the sizes of $|A|$ and $|B|$ do not matter).

Example:

$$
\begin{aligned}
A &= \{1, 4\} \\
B &= \{1, 3\} \\
\\
C &= \{2, 4, 5, 7\}
\end{aligned}
$$

Exercise 9-3. Do Problem 35-5, on page 1051 of CLRS.