University of British Columbia
Computer Science Department

MIDTERM EXAMINATION IN CS 500

SPRING 2009

Professor William Evans

This is a closed book examination. You have 1.5 hours. Do all 4 problems. Write your solutions on the examination sheets.

Your name: _____

1. (25 points) **Card Shuffling** Consider the following (not very efficient) process for card shuffling. Start with a deck of n cards in sorted order. Repeatedly, take the top card and place it at a randomly chosen position in the deck. (Each of the n positions is equally likely.) Stop after the original bottom card appears on top and is placed in a randomly chosen position.

   Show that the expected number of operations until the process is completed is $O(n \log n)$.

   Hint: Consider the times at which the original bottom card changes position.

2. (25 points) **Zero sum subset** Describe an efficient dynamic programming algorithm that given a set of integers $S = \{x_1, x_2, \ldots, x_n\}$ finds a non-empty subset R of S so that

$$\sum_{x \in R} x = 0 \pmod{n}.$$

Hint: Consider the more general problem of what values $\pmod{n}$ can be obtained.

You do not need to write code, just explain how to fill in the dynamic programming table. What is the running time of your solution?

3. (25 points) **String Matching with Wildcards** Let $* \notin \Sigma$ be a *wildcard* character. A wildcard character in a pattern matches any *string* of characters from Σ. For example, the pattern `ab*b*b` matches `abbb` and `ababaab` but not `ababa`. Given a pattern $p = p_1 p_2 \dots p_m$ where $p_i \in \Sigma \cup \{*\}$, explain how you would compute a modified version of the Knuth-Morris-Pratt string matching automaton for p. The construction of your automaton should take $O(m)$ time, and it should detect the first occurrence of p in a text $t = t_1 t_2 \dots t_n$ ($t_i \in \Sigma$) in $O(n)$ time.

4. (25 points) **Weighted Median** For n distinct elements $x_1, x_2, \ldots, x_n$ with positive weights $w_1, w_2, \ldots, w_n$ such that $\sum_{i=1}^{n} w_i = 1$, the *weighted median* is the element x_k satisfying

$$\sum_{x_i < x_k} w_i < \frac{1}{2} \qquad \text{and} \qquad \sum_{x_i > x_k} w_i \leq \frac{1}{2}.$$

Describe how to compute the weighted median of n elements in $O(n)$ worst-case time.

Hint: Use the deterministic linear-time median algorithm that we discussed in class (you do not need to describe that algorithm) to construct an efficient recursive algorithm to find the weighted median.