CS 337	**Test 3**	**5/12/08**

Open book and notes.

Max points = 75 Time = 75 min Do all questions.

1. (Recursion and Induction; 45 points)

   (a) (4 points) Define function `take` where `take n xs` is a list containing the first `n` items of list `xs` in order, where `n` is a natural number. If `n` exceeds the length of `xs` then the entire list `xs` is returned.

   (b) (4 points) Define function `drop` where

   ```
   xs = (take n xs) ++ (drop n xs)
   ```

   Recall: `++` concatenates two lists.

   (c) (9 points) A matrix of integers is represented as a list of lists where each component list is a row of the matrix. Thus, the matrix in Table 1

3	7	4
2	1	9
1	2	4
0	6	2

   Table 1: Matrix

   is represented by $[[3, 7, 4], [2, 1, 9], [1, 2, 4], [0, 6, 2]]$. Assume that the matrix is not empty, so no row or column is empty.

   Write functions to compute: (1) a list of row sums, $[14, 12, 7, 8]$ for this example, (2) the sum of all the matrix elements, 41 in this case, and (3) a list of column sums, $[6, 16, 19]$ in this case.

   You may have to write auxiliary functions. Use *map* and *foldr* liberally.

   (d) (10 points) You are asked to find the i^{th} value in rank order in a given list of n integers, $n > 0$. Assume that $1 \leq i \leq n$. For example, given the list [31,34,22,3,19,5,20,6], its sorted version is [3,5,6,19,20,22,31,34]. For $i = 1$, the value is 3, for $i = 2$, the value is 5, for $i = 5$, the value is 20, etc.

   To find the i^{th} value, you don't need to sort the list. You can use function `partition`, given in Page 151 in the class notes. Given a list `(x:xs)`, partition `xs` into lh and rh using `x` for `v` in function `partition`. If the length of lh is greater than or equal to i, look for the i^{th} value in lh, otherwise, find the suitable value as x or in rh.

   Define function `rank i xs`. Use function `partition` (you don't have to rewrite it). You can get the length of a list using Haskell function `length`. Assume that the list elements are distinct.

(e) (9 points) You are given a list `xs` of strings. Write a function to produce a list of strings `ys`, of the same length as `xs`, where the i^{th} string of `ys` is the concatenation of all strings of `xs` up to and including the i^{th} one. Thus, given that `xs` is `[''this'', ''is'', ''cute'']`, `ys` is `[''this'', ''thisis'',''thisiscute'']`. Use `++` for list (string) concatenation.

(f) (9 points) You are given a list `xs` of bit strings, all of the same length. Write a function to output `True` iff all adjacent pairs of strings have Hamming distance of 1. Consider the first and last strings of `xs` to be adjacent. Thus, `["00", "01", "11", "10"]` should produce `True` and `["00", "01", "10", "11"]` produces `False`. For input `[]`, output `True`.

Hint: You may use the function `right_rotate` that rotates a list to the right by one position. This function is defined in Page 145 of the notes. Just use the function, don't reproduce it.

2. (String Searching; 20 points)

(a) (Core Computation; 8 points) Let Z_i be the alternating binary string $\overbrace{0101\cdots01}^{2i}$ of length $2i$ (i.e., i pairs of 01). Thus, Z_4 is 01010101. Show that $c^j(Z_n0) = Z_{n-j}0$, where c denotes the core function, and $1 \le j \le n$. That is, $c^1(Z_40) = c(010101010) = 0101010 = Z_30$, and $c^3(Z_40) = c^2(c(010101010)) = c^2(0101010) = c(c(0101010)) = c(01010) = 010 = Z_10$.

(b) (Core Computation; 6 points) Consider string $p[0..12]$ where $p[3]$, $p[5]$ and $p[9]$ are all different symbols. What can you say about the length of the core?

(c) (Rabin-Karp algorithm; 6 points) We are given a very long genome sequence. This string will be repeatedly searched with patterns of small lengths, say of lengths about 20. How can we apply the Rabin-Karp algorithm effectively? There are several possible approaches; any good one will do. Explain your algorithm in no more than 5 sentences.

3. (Relational Algebra; 10 points)

(a) (5 points) Let the Tables 6.1 (page 170), 6.4 (page 172) and 6.5 (page 173) in your notes be denoted by relations R, S and T, respectively. Write a query to find the names of theatres which are showing PG movies in which Will Smith is acting. You don't have to simplify or compute the value of the query.

(b) (5 points) Show that for arbitrary relations R, S , and attribute a, $\pi_a(R \cap S) = \pi_a(R) \cap \pi_a(S)$ does not necessarily hold.