Homework on String Algorithms

Write the solution to each problem on a single page of a separate sheet of paper.
The deadline for handing in solutions is November 29.

Problem 1. (20 points). The cyclic rotation by $k$ positions transforms a string $T = a_0, a_1, \ldots, a_{n-1}$ into another string $T_k = a_k, \ldots, a_{n-1}, a_0, \ldots, a_{k-1}$. Assuming $T$ is stored in a linear array of length $n$, give an in-place algorithm that constructs $T_k$ in time $O(n)$, independent of $k$. [To be in-place, the algorithm can only use a constant amount of extra memory, so no second array for temporarily storing portions of $T$ is permitted.]

Problem 2. (20 points). Give a linear-time algorithm to determine whether or not a string $T_1$ is a cyclic rotation of another string $T_2$. For example, $a r c$ is a cyclic rotation of $c a r$.

Problem 3. (20 points). Take a constant size alphabet and consider a cyclic string $a_0, a_1, \ldots, a_{n-1}$ (defined by letting $a_{i+1 \mod n}$ be the successor of $a_i \mod n$, for each $i$) over this alphabet. If we cut the cycle between positions $i$ and $i + 1$, we get a linear string $a_i, a_{i+1}, \ldots, a_{n-1}, a_0, \ldots, a_i$. Give a linear-time algorithm for finding the cut that produces the lexicographically smallest linear string. [Your solution may use a linear-time algorithm for constructing the suffix tree of a string as a black box.]

Problem 4. (20 = 6 + 8 + 6 points). Remember that $I, V, X,$ and $L$ are the symbols for 1, 5, 10, and 50 in roman numbers.

(a) Describe the rules for constructing roman numbers between 1 and 89 in words.

(b) Give a corresponding regular expression $A$.

(c) Draw the directed graph whose language is the set of strings defined by $A$.

Problem 5. (20 = 10 + 10 points). How would you express the following two additional symbols in the specification of a regular expression?

(a) The wild-card, *, that indicates a position in the pattern that can be occupied by any character.

(b) The negation, $\bar{a}$, that prevents a position to be occupied by $a$ but any other character is allowed.