**58093 String Processing Algorithms (Autumn 2012)**
Exercises 4 (22 November)

1. Two string $x$ and $y$ are *rotations* of each other if there exists strings $u$ and $v$ such that $x = uv$ and $y = vu$. For example `abcde` and `deabc` are rotations of each other. Describe a *linear time* algorithm for determining whether given two strings are rotations of each other. (Hint: use a linear time exact string matching algorithm.)

2. The Knuth–Morris–Pratt algorithm differs from the Morris–Pratt algorithm only in the failure function, which can be defined as

   $fail_{\text{KMP}}[i] = k$, where $k$ is the length of the longest proper border of $P[0..i)$ *such that* $P[k] \neq P[i]$, or $-1$ if there is no such border.

   (a) Compute both failure functions for the pattern `ananassana`.

   (b) Give an example of a text, where some text character is compared three times by the MP algorithm but only once by the KMP algorithm when searching for `ananassana`.

3. Modify Algorithm 2.6 on the lecture notes to compute $fail_{\text{KMP}}$ instead of $fail_{\text{MP}}$.

4. A don't care character # is a special character that matches any single character. For example, the pattern `#oke#i` matches `sokeri`, `pokeri` and `tokeni`.

   (a) Modify the Shift-And algorithm to handle don't care characters.

   (b) It may appear that the Morris–Pratt algorithm can handle don't care characters almost without change: Just make sure that the character comparisons are performed correctly when don't care characters are involved. However, such an algorithm would be incorrect. Give an example demonstrating this.

5. Simulate the execution of the BNDM algorithm for the pattern `anna` and the text `bananamanna`.

6. Let $\mathcal{P} = \{P_1, \ldots, P_{2k}\}$ be a set of patterns such that

   - for $i \in [1..k]$, $P_i = \mathtt{a}^i$ and
   - for $i \in [k+1..2k]$, $P_i = P_i'\mathtt{a}^k$ such that $|P_i'| = k$ and each $P_i'$ is different.

   (a) Show that the total size of the sets *patterns*$(\cdot)$ in the Aho–Corasick automaton for $\mathcal{P}$ is asymptotically larger than $||\mathcal{P}||$.

   (b) Describe how to represent the sets *patterns*$(\cdot)$ so that

   - the total space complexity is never more than $||\mathcal{P}||$
   - each set *patterns*$(\cdot)$ can be listed in linear time in its size.