58093 String Processing Algorithms (Autumn 2013)
Exercises 4 (19 November)

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

2. Show how the following (single) exact string matching algorithms can be modified to solve the *multiple exact string matching problem*:

   (a) Shift-And

   (b) Karp-Rabin

   The solution should be more efficient than the trivial one of searching each pattern separately.

3. Given a text T and pattern P, the *longest prefix matching problem* is to find the longest prefix of the pattern that occurs in the text as a factor.

   (a) Show how to modify the (K)MP algorithm to solve this problem.

   (b) Which other algorithms from the lectures could be easily modified to solve this problem?

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. Prove the following Weak Periodicity Lemma

   If p and q are periods of S and $p + q \leq |S|$, then $gcd(p, q)$ (greatest common divisor) is a period of S too.

   or even the following Strong Periodicity Lemma

   If p and q are periods of S and $p + q - gcd(p, q) \leq |S|$, then $gcd(p, q)$ is a period of S too.

6. Let $\mathcal{P}_k = \{P_1, \ldots, P_{2k}\}$ 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 $\mathcal{P}_k$ is asymptotically larger than $||\mathcal{P}_k||$.

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

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