58093 String Processing Algorithms (Autumn 2010)
Exercises 2 (18 November)

1. The multiple exact string matching problem is to find the occurrences of multiple patterns $P_1, P_2, \ldots, P_k$ in a text T. The trivial solution is to find each pattern separately. Show how the following algorithms can be modified to solve the problem more efficiently:

   (a) Shift-And

   (b) Karp-Rabin

2. 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`. Modify the following exact string matching algorithms to handle the case where the pattern may contain don't care characters.

   (a) Shift-And

   (b) Horspool

   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.

   (c) Give an example demonstrating this.

3. Show that edit distance is a *metric*, i.e., that it satisfies the metric axioms:

   - $ed(A, B) \geq 0$
   - $ed(A, B) = 0$ if and only if $A = B$
   - $ed(A, B) = ed(B, A)$ (symmetry)
   - $ed(A, C) \leq ed(A, B) + ed(B, C)$ (triangle inequality)

4. Describe a family of string pairs (A_i, B_i), $i = \mathbb{N}$, such that $|A_i| = |B_i| \geq i$ and there is at least i different optimal edit sequences corresponding to $ed(A_i, B_i)$. Can you find a family, where the number of edit sequences grows much faster than the lengths of the strings?

5. Let $P =$ `evete` and $T =$ `neeteneeveteen`.

   (a) Use Ukkonen's cut-off algorithm to find the occurrences of P in T.

   (b) Simulate the operation of Myer's bitparallel algorithm when it computes column 5 for P and T.

6. Give a proof for Lemma 2.15 in the lecture notes.