CSMTSP Text Searching Processing

1. **(a)** Design the Aho-Corasick (AC) automaton over the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ for the dictionary of strings: `aaababa`, `aababa`, `aba`, `bab`. *[15 marks]*

(b) Define and explain the data structure used to implement an AC automaton for dictionary matching. *[15 marks]*

(c) Describe in pseudo-code the search procedure for finding all the occurrences of the patterns in the dictionary on a text using the implementation described in (b). *[10 marks]*

(d) State the time and space complexity for building the AC automaton of a set of strings X, as well as the complexity of search procedure applied to a text of length n. State the maximal time spent by the search procedure on a single symbol of the text. *[10 marks]*

2. **(a)** Give an example of the "match shift" (good-suffix rule) used by the Boyer-Moore (BM) string matching algorithm. Give the definition of the match shift table D. *[15 marks]*

(b) Give an example of the "occurrence shift" (bad character rule) used by the BM string matching algorithm. Describe in pseudo-code the procedure for building the occurrence shift table DA. *[15 marks]*

(c) Describe in pseudo-code the search procedure of the BM string matching algorithm. State it's "worst-case" and "best case" time complexity backed up with examples. *[10 marks]*

(d) Recall the definition of R_next used in the computation of the table D of part (a). Let x be a pattern of length m then $R\_next[j] = m - |Border(x[j+1 \dots m])|$ for all $0 \leq j \leq m$. Describe in pseudo-code the construction of the table D to implement the match shift (good suffix rule) assuming you are given the array R_next. *[10 marks]*

See Next Page

3. In the following we only consider the binary alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$.

(a) Describe in pseudo-code the construction of the *KMP_next* array used in the Knuth-Morris-Pratt (KMP) string-matching algorithm. *[10 marks]*

(b) Give the *KMP_next* array for the pattern $x =$ `ababbabaab`. *[10 marks]*

(c) Describe in pseudo-code the search procedure of the KMP string-matching algorithm. *[10 marks]*

(d) For $x \in \Sigma^*$, the string-matching automaton of x, $SMA(x)$, is the minimal deterministic automaton accepting the language $\Sigma^* x$. Design the following string-matching automaton $SMA($`ababbabaab`$)$. *[10 marks]*

(e) State the maximal time spent by the KMP search procedure on a single letter of a pattern y of length m on a text over a two-letter alphabet and over a three-letter alphabet. *[10 marks]*

4. **(a)** Define the Hamming and Levenshtein distances of two strings x and y followed by an example. *[10 marks]*

(b) Write an algorithm that computes the edit distance matrix C for two strings x and y, where $|x| = n$, $|y| = m$, and $C[i, j]$ is the cost of a cheapest edit script that transforms the first i characters of x into the first j characters of y. *[10 marks]*

(c) Expand the algorithm of part (b) so that it produces and outputs one optimum edit script. That is, it should recover a sequence of edit operations (insert, delete, substitute) that results in a minimum total cost. *[15 marks]*

(d) Prove that when insertion and deletion have unit cost and the cost of a nontrivial substitution (*i.e.*, a substitution in which a character is replaced by a different one) is at least 2, then the minimum edit distance e between two strings of length m and n is $e = n + m - 2s$, where s is the length of a longest common subsequence between x and y. *[15 marks]*

See Next Page

5. **(a)** Construct the suffix tree for the string `aabaabba`. *[10 marks]*

(b) Define and explain the data structures used to implement a suffix tree *[15 marks]*

(c) Let S_y be the suffix link funciton of the suffix tree of y. Prove that $S_y(p)$ is a fork if p is a fork in the tree. Give the lower and upper tight bounds on the number of nodes in the tree. *[15 marks]*

(d) Describe in pseudo-code the computation of $S_y(p)$. Give an outline in your own words of an algorithm to compute $S_y(p)$ if it is the only one not yet defined in the structure. *[10 marks]*