—— SOLUTIONS ——

King's College London

UNIVERSITY OF LONDON

This paper is part of an examination of the College counting towards the award of a degree. Examinations are governed by the College Regulations under the authority of the Academic Board.

MSc & MSci EXAMINATION

7CCSMTSP – TEXT SEARCHING AND PROCESSING

MAY 2012

TIME ALLOWED: TWO HOURS.

ANSWER TWO OF THE THREE QUESTIONS.

NO CREDIT WILL BE GIVEN FOR ATTEMPTING ANY FURTHER QUESTIONS.

ALL QUESTIONS CARRY EQUAL MARKS.

THE USE OF ELECTRONIC CALCULATORS IS **NOT** PERMITTED.

BOOKS, NOTES OR OTHER WRITTEN MATERIAL MAY **NOT** BE BROUGHT INTO THIS EXAMINATION.

DO NOT REMOVE THIS EXAM PAPER FROM THE EXAMINATION ROOM

TURN OVER WHEN INSTRUCTED

1. String Matching

a. Assume we are given a single pattern. Name an algorithm that can be used to efficiently search for the pattern in many texts? What are the possible preprocessing time for a pattern of length m and the running time for searching k texts each of length n?

Assume we are given a single text. Name an algorithm that can be used to search the text for many patterns? What are the possible preprocessing time for a text of length n and the running time to search for k patterns each of length m?

[10 marks]

Answer

In the first case the pattern can be preprocessed once for searching all texts. Then MP algorithm can be applied. The preprocessing time can be done in $O(m)$ time and the whole search in $O(kn)$ time. [5 marks]

In the second case the text is to be preprocessed (indexed), building its Suffix tree, Suffix array or Suffix automaton. The preprocessing time can be done in $O(n)$ or $O(n \log a)$ time (a is the alphabet size) and the whole search in $O(km)$ or $O(km \log a)$ time depending on the representation of the index. [5 marks]

[unseen]

b. Let x be the string ababaabab. Give its Border table B ($B[i]$ is the maximal length of borders of $x[0 \,.\,. i-1]$), its Period table P ($P[i]$ is the smallest period of $x[0 \,.\,. i-1]$), and its *MP_next* table.

[10 marks]

Answer

	0	1	2	3	4	5	6	7	8	9
	a	b	a	b	a	a	b	a	b	
B	-1	0	0	1	2	3	1	2	3	4
P	–	1	2	2	2	2	5	5	5	5
MP_next	-1	0	0	1	2	3	1	2	3	4

[3 marks for each row] [unseen]

c. Design and describe in pseudo-code an algorithm that computes the $Border$ table of a word x of length m and runs in time $O(m)$.

[10 marks]

<u>Answer</u>

```
COMPUTE_BORDERS(string x; integer m)
    Border[0] ⟵ −1
    FOR(i ⟵ 0 to m − 1)
        j ⟵ Border[i]
        WHILE(j ≥ 0 and x[i] ≠ x[j])
            j ⟵ Border[j]
        Border[i + 1] ⟵ j + 1
RETURN Border
```

[bookwork]

d. A string w is called a cover of x if it is a prefix of x and if any two consecutive positions, i and j, $i < j$, of w on x satisfy $j - i \leq |w|$. (Note that the second occurrence of w at position j may be an overhanging occurrence.) For example, aba is the shortest cover of ababaabab.

Given two strings w and x, design in your own words how to test whether w is a cover of x. What is the running time of your algorithm?

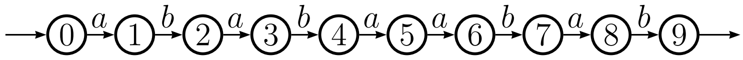
[10 marks]

<u>Answer</u>

Testing if w is a prefix of x is straightforward and takes time $O(|x|)$. If it is, we then compute all the positions of w on x. With MP algorithm this is done in time $O(|w| + |x|)$, which is $O(|x|)$ because w is a prefix of x, and we get them in increasing order. Finally we check that the distance between any two consecutive positions is at most $|w|$. This is done again in time $O(|x|)$.

[unseen]

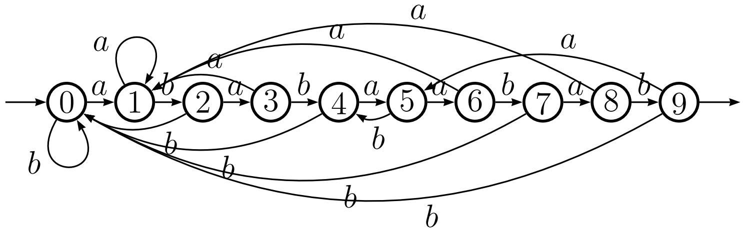
e. What is SMA(x), the String Matching Automaton of the string x? Extend the following automaton into SMA(ababaabab) on the alphabet $A = \{a, b\}$.



[10 marks]

<u>Answer</u>

SMA(x) is the smallest deterministic automaton accepting all the strings that end with x.



[unseen]

2. **Searching a list of strings**
Consider a list of strings $L = (y_1, y_2, \ldots, y_k)$ in lexicographic order: $y_1 \leq y_2 \leq \cdots \leq y_k$. Let x be a string that is to be searched for in the list L. All strings x and y's have the same length.

a. What is the asymptotic running time of a binary search for x in L if no extra information on the strings y's is known? Give a "worst-case" example to your answer.

[10 marks]

Answer

The asymptotic cost of a binary search for the string x of length n in the list L of k lexicographically sorted strings y_i is $O(n \log k)$ time. A "worst-case" example could be the search for $x = bbb \ldots b$ in the list $L = (aaa \ldots a, aaa \ldots b, aaa \ldots bb, aaa \ldots bbb, \ldots, bbb \ldots b)$

b. For two strings u and v, $\mathrm{lcp}(u, v)$ denotes the maximum length of their common prefixes. Let $\ell = \mathrm{lcp}(x, y_1)$, $r = \mathrm{lcp}(x, y_k)$, and $i = \lfloor (k+1)/2 \rfloor$. Assume that $y_1 \leq x \leq y_k$ and $\ell > r$. How does x compare with y_i when $\ell < lcp(y_1, y_i)$? How does x compare with y_i when $\ell > lcp(y_1, y_i)$?

[10 marks]

Answer

Assume $l < lcp(y_1, y_i)$. Then let $u = y_1[1 \ldots l]$, $\sigma = y_1[l + 1]$, $\tau = y_i[l + 1]$. Then $u\tau$ is a prefix of x and $\sigma < \tau$. This implies that $y_i < x < y_k$.
Now assume $l > lcp(y_1, y_i)$. In this case, we have that $\sigma \neq \tau$ and $\sigma < \tau$ which implies that $y_1 < x < y_i$.

c. Give the Suffix Array of the string `ababaabab`.

[10 marks]

Answer

i	0	1	2	3	4	5	6	7	8
$y[i]$	a	b	a	b	a	a	b	a	b
SUF[i]	4	7	2	5	0	8	3	6	1
LCP[i]	0	1	2	3	4	0	1	2	3

d. State the running time of the search for occurrences of x in y using the Suffix Array of y.

[10 marks]

Answer

Running a binary search for a string x of length m by using the longest common prefixes of the n sorted suffixes of y takes $O(m + \log n)$ time.

e. State the running time for computing the LCP table for a string of length n. In which order are suffixes to be considered?

[10 marks]

Answer

The computation runs in linear by considering the suffixes from the longest to the shortest.

3. **Suffix structures**

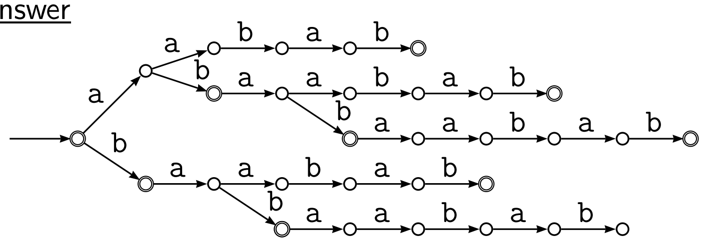
In this question we consider the string $z = \texttt{ababaabab}$.

a. Design the Suffix trie of the word z.

Give an example of a word of length n on the alphabet $\{\texttt{a}, \texttt{b}\}$ that has a Suffix trie of size $\Omega(n^2)$.

[10 marks]

<u>Answer</u>



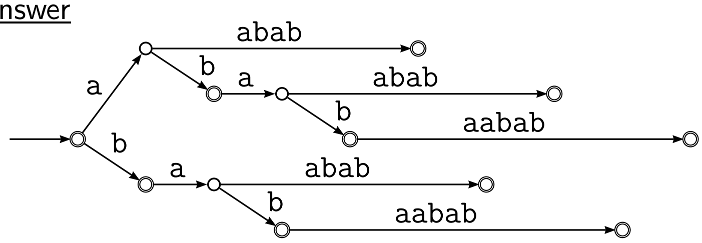
[5 marks] [unseen]

The trie of the word $a^{n/4}b^{n/4}a^{n/4}b^{n/4}$, for two distinct letters a and b, has at least $n/4$ branches each of them having $n/4$ nodes. Which gives at least $(n/4)^2 = \Omega(n^2)$ nodes. [5 marks] [in lectures]

b. Design the Suffix tree of z. What are the properties characterising the Suffix tree of a non-empty string y? Describe how to get the Suffix tree of y from its Suffix trie.

[10 marks]

Answer



[5 marks] [unseen]

The Suffix tree of y is a digital tree in which edges are labelled by non-empty strings. The label of a path from the root to a terminal node is a suffix of y and all the suffixes of y appear as such. Edges outgoing a given node have labels starting with different letters. No node has only on outgoing edge. [bookwork]

Each node having only one outgoing edge in the Suffix trie should be deleted to get the Suffix tree, and edges should be labelled accordingly. For edges (p, u, q) and (q, a, r) where u is a word, a a letter and q has no other outgoing edge, it is deleted with the two edges, and the new edge (p, ua, r) is created. [5 marks] [unseen]

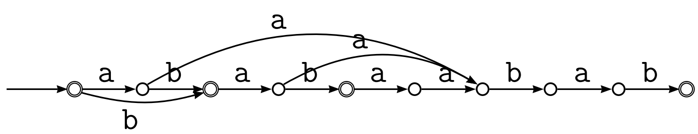
c. What is the Suffix automaton of a non-empty string y? Describe in your own words how to get the Suffix Automaton of y from its Suffix Trie. Design the Suffix automaton of the string z.

[10 marks]

Answer

The Suffix automaton of y is the minimal deterministic automaton accepting the suffixes of y.

The Suffix automaton of y can be obtained by minimising its Suffix trie. [5 marks]



[5 marks] [unseen] [bookwork]

d. How would you implement nodes and edges of a Suffix tree? Describe how to discover if a pattern x of length m occurs in a string y using the Suffix tree of y. Discuss the running time of the method with respect to your implementation of the tree.

[10 marks]

Answer

To discovering if x occurs in y we just have to follow the path labelled by x in the Suffix Tree. If the path does not exist, x does not occur in y. Otherwise it occurs. [5 marks]

If the tree is represented by a transition (goto) table, each branching from a state takes constant time, which leads to $O(m)$ time. If the tree is represented by successor lists it takes $O(m \log a)$, where a is the number of letter in the alphabet, because the tree is deterministic and then there are no more than a edges outgoing a given state; the edges can be arranged in a balanced tree to get $O(\log a)$ for branching. [5 marks]

[mostly unseen]

e. Describe in your own words how to find a longest factor of y occurring at least twice in it, using the Suffix tree of y. Answer the same question using the Suffix automaton of y.

[10 marks]

Answer

We have to find a lowest internal node in the tree. Since it is internal, the label from the root to it occurs twice in y. [5 marks]

We can process the automaton by marking states leading to at least two terminal states. The marked state having the maximal L value is an answer. [5 marks] [unseen]