

— 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.

BSc EXAMINATION

6CCS3TSP – 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

[4 marks for first row, 3 for next two rows] [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  $O(m)$ .

[15 marks]

Answer

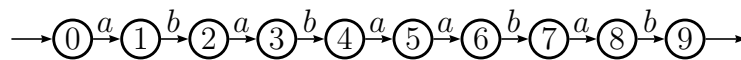
```

COMPUTE_BORDER(string  $x$ ; integer  $m$ )
  Border[0]  $\leftarrow$  -1
  FOR( $i \leftarrow 0$  to  $m - 1$ )
     $j \leftarrow$  Border[ $i$ ]
    WHILE( $j \geq 0$  and  $x[i] \neq x[j]$ )
       $j \leftarrow$  Border[ $j$ ]
    Border[ $i + 1$ ]  $\leftarrow$   $j + 1$ 
  RETURN Border

```

[bookwork] [5 marks for the idea, 10 more for correct code]

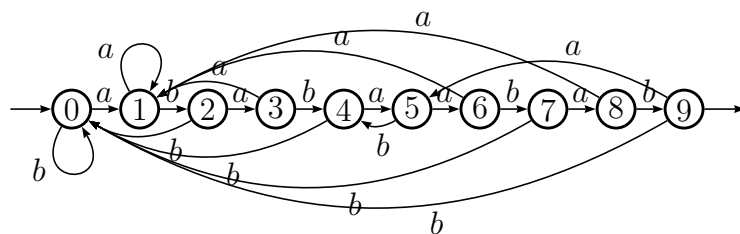
- d. What is  $SMA(x)$ , the String Matching Automaton of the string  $x$ ? Extend the following automaton into  $SMA(ababaabab)$  on the alphabet  $A = \{a, b\}$ .



[15 marks]

Answer

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



[unseen] [1 mark for each correct extra edge, 15 marks if totally correct]

## 2. Searching a list of strings

Consider a list of strings  $L = (y_1, y_2, \dots, y_k)$  in lexicographic order:  $y_1 \leq y_2 \leq \dots \leq y_k$ . 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  $x = bbb \dots b$  in the list  $L = (aaa \dots a, aaa \dots b, aaa \dots bb, aaa \dots bbb, \dots, bbb \dots b)$

[bookwork] [10 marks]

- b. Give the Suffix Array of the string ababaabab.

[15 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

[unseen] [8 marks for first row, 7 marks for second row]

- c. State the running time of the search for occurrences of  $x$  in  $y$  using the Suffix Array of  $y$ .

[15 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.

[bookwork] [8 marks for  $\log n$ , 7 marks for  $m$ ]

- d. 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.

[bookwork] [10 marks]

### 3. Suffix structures

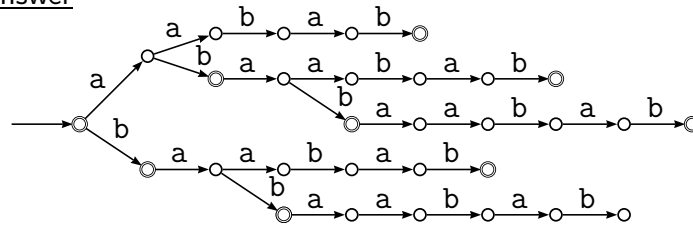
In this question we consider the string  $z = \text{ababaabab}$ .

- a. Design the Suffix trie of the word  $z$ .

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

[10 marks]

Answer



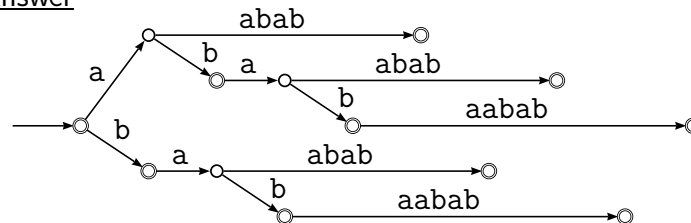
[5 marks] [unseen]

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.

[15 marks]

Answer



[8 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 one 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. [7 marks]

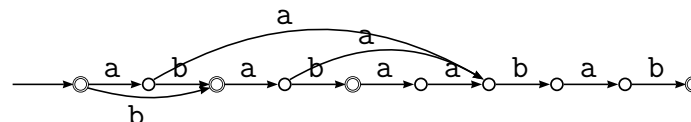
[unseen]

- 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  $z$ .

[15 marks]

Answer

The Suffix automaton of  $y$  is the minimal deterministic automaton accepting the suffixes of  $y$ . [5 marks] [bookwork]



[5 marks] [unseen]

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