THE UNIVERSITY of LIVERPOOL

May 2006 EXAMINATIONS

Model solutions

Applied Algorithmics

TIME ALLOWED : 2.5 hours

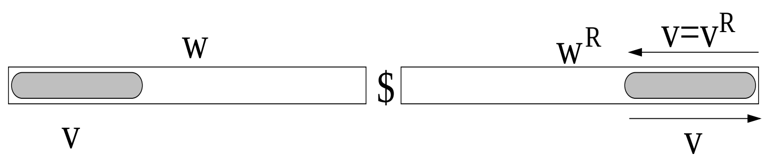
INSTRUCTIONS TO CANDIDATES

Candidates will be assessed on their best four answers. If you attempt to answer more than the required number of questions, the marks awarded for the excess questions will be discarded (starting with your lowest mark).

THE UNIVERSITY
of LIVERPOOL

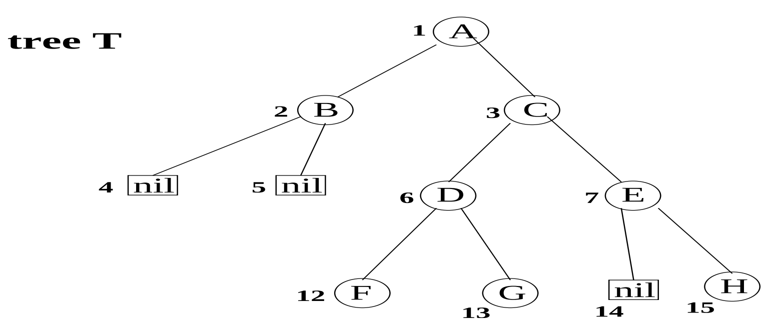
Answers to question 1

1.A The KMP failure function computed for a finite string string $w = w[0..n-1]$ is represented as a vector F[1..n], s.t., each F[i] contains the length of the longest proper prefix which is also a suffix of prefix $w[0..i-1]$, for all $i = 1,..,n$. In order to find the longest prefix of w which is also a palindrome we can compute the failure function for the word $z = w\$w^R$, where \$ is a special character which does not occur in w.
Note that any prefix of z longer than n cannot be a suffix of z due to the difficulty with the central symbol \$. Any shorter prefix of z must be also a prefix of w. Moreover, any prefix of w which is a suffix of z, must be a palindrome, see figure below. Thus the value of the failure function $F[2n+1]$ computed on $z = w\$w^R$ contains information about the longest prefix of w which is also a palindrome. The time complexity of the solution is $O(n)$, and it corresponds to the time required to compute the failure function for string w.



1.B Dense binary trees can be efficiently represented in the form of an array/vector $V[1..]$. And, for a given binary tree T the root r of the tree is represented by the the first element of vector $V[1]$. And for any internal node v (including the root), represented by entry $V[i]$ of the tree T the left child of v is available in $V[2 \cdot i]$ and the right child is available in $V[2i+1]$. The vector representation of binary trees is used, e.g., in *Heaps*, one of the most efficient implementation of priority queues.

The vector $V[1..15]$ shown in question 1.B represents the following tree.

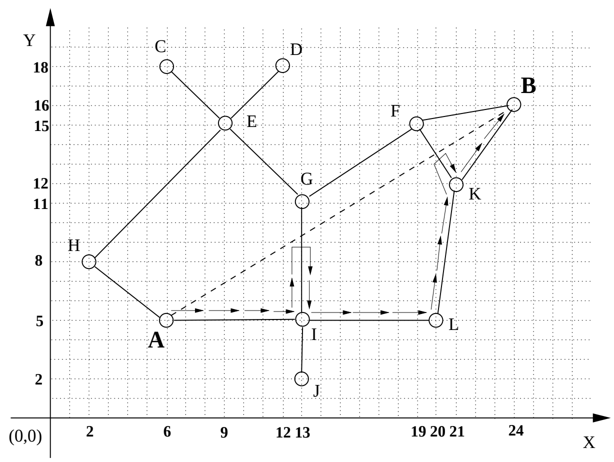


1.C The three most important basic properties of the abstract radio network protocol are:

1. During any time step a network node can either transmit or receive messages;
2. When a node v transmits a message it is transmitted to all its neighbours in the underlying graph of connections;
3. During any time step a node can receive a message iff only one of its negihbours is transmitting at this time step.

Answers to question 2

2.A A planar (and embedded into plane) graph $G = (V, E)$ possess Gabriel graph property if any two points $x, y \in V$ are connected iff the disk (circle) having segment (x, y) as a diameter contains no other points from V. In the view of this definition one can eliminate: edge (C, D) (disk contains point E), edge (A, G) (disk contains point I), and edge (I, F) (disk contains point G).
The route from A to B generated by *Compass Routing* algorithm is:



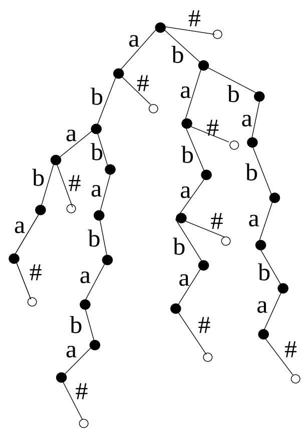
2.B MTF algorithm traverses through an input sequence value by value and replaces each value by its current position in the list and at the same time it moves this value to the beginning of the list. The MTF procedure is use to decrease the size of the input sequence (potentially large values are replaced by smaller dynamic indexes in the list). In case of input sequence S and initial content of Q the transformation process is as follows:

0)	**[10,50,60]**	1	5)	$[60, 10, 50]$	3
1)	$[10, 50, 60]$	2	6)	$[50, 60, 10]$	3
2)	$[50, 10, 60]$	1	7)	$[10, 50, 60]$	1
3)	$[50, 10, 60]$	2	8)	$[50, 10, 60]$	1
4)	$[10, 50, 60]$	3	9)	**[10,50,60]**	

2.C *Lowest Common Ancestor* (LCA) for two nodes u and v in a rooted (in root r) tree T is the lowest node in T that belongs both to path from v to r and from u to r. The LCA problem is to initially process the the rooted tree, s.t., the further LCA queries are as cheap as possible. In particular it is known how to process a tree of size n in time $O(n)$, s.t., further LCA queries are served in time $O(1)$.

3.A The suffix tree and suffix array for string *abbababa* are as follows.

Suffix tree



String: a b b a b a b a
0 1 2 3 4 5 6 7

Suffix array

7	[a]
5	[aba]
3	[ababa]
0	[abbababa]
6	[ba]
4	[baba]
2	[bababa]
1	[bbababa]

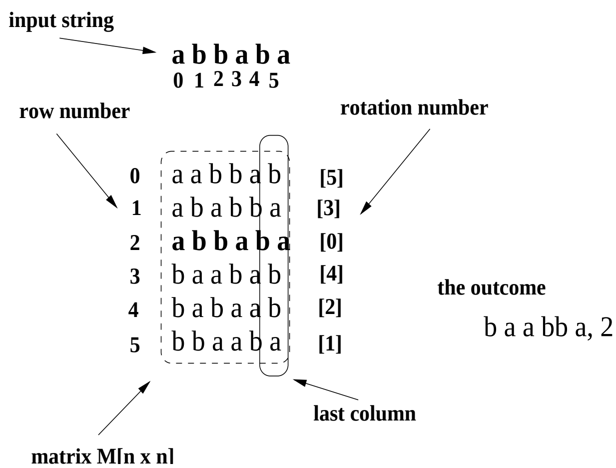
Suffix arrays are always linear $O(n)$ in the size n of an input string. Standard suffix tree might be as large as $\Omega(n^2)$, e.g., when all symbols in the input string are different. However, one can construct a compact suffix tree of size $O(n)$, which is a standard suffix tree in which all chains are replace by single reference to the appropriate substring.

3.B The initial strings in Babanacci Language are: $B_0 = ab$, $B_1 = ba$, $B_2 = baab$, $B_3 = baabba$, and $B_4 = baabbabaab$. A string $W = W[0..n-1]$ has a period p iff $W[i] = W[i+p]$, for all $i = 0,..,n-p-1$. If this is not the case, i.e., p is not a period of W, there is a *witness* against non-period p, and it is defined as any position $i \in \{p,..,n\}$, s.t., $W[i] \neq W[i-p]$. And in particular in case of string $B_4[0..9] = baabbabaab$ we can take position 7 as a witness against non-period 4, since $B_4[7] \neq B_4[3]$.

3.C For any two strings u and v of the same length n, the *Hamming distance* $HD(u,v)$ is the number of positions on which two strings differ, and the *relative Hamming distance* $RHD(u,v)$ is defined as a fraction of positions on which two strings differ, i.e., $RHD(u,v) = \frac{HD(u,v)}{n}$.

THE UNIVERSITY *of* LIVERPOOL

4.A The Burrows-Wheeler transform takes all cyclic rotations on an input string of length n, sorts them sort them according to lexicographical order and place as rows in a $[n \times n]$ matrix M. As the outcome of the transformation we take the last column in the matrix M and the number of a row that contains the original string. See the picture below.



4.B The three diagrams A), B) and C) in question 4.B illustrate shifting mechanisms of the Knuth-Morris-Pratt (KMP) algorithm, Boyer-Moore (BM) algorithm and Karp-Rabin (KB) algorithm.

The shifting mechanism in the KMP algorithm (case A) is based on precomputation of all prefixes of the pattern searched in the text. In particular, for each prefix p of the pattern we precompute its longest proper prefix x which is also the suffix of p. After encounter of a mismatch between the pattern and the text symbol we perform the shift of size $|p| - |x|$.

The shifting mechanism in the BM algorithm (case B) is based on recognition of the pattern suffixes. In particular, for each pattern suffix starting at position i we precompute its earlier (but the rightmost) occurrence in the pattern j. After encounter of a mismatch between the pattern and the text symbol we perform shift of size $i - j$. Typically the shifts in the BM algorithm are longer than in the KMP algorithm.

The last case C) reflects execution of the shifting mechanism in the KB algorithm, where the shift is always of length 1. The pattern is compared with the text via application of easily computable hashing function, which allows to recompute its value after removal of one symbol in the beginning and inclusion of one symbol at the rear part of the tested fragment of the text. The shifting mechanism in the KB algorithm is very simple though it allows to perform various (e.g., approximate) variants of the text search, while the other two shifting mechanisms work only in the exact string matching.

4.C The decreasing function method is defined as a support tool that helps to decide whether a given loop (e.g., while, repeat, etc) finishes its execution in predictable time. The decreasing function is a potential integer function with bounded (limited) original value. One has to show that after each iteration of the body of the loop the value of the function becomes strictly smaller but it never reaches 0 (or any negative value). This is enough to prove that the loop eventually finishes its execution.

THE UNIVERSITY of LIVERPOOL

5.A A binary matrix M is a $(k,k,n)-$selector iff for any choice of k columns in M there are some k rows, s.t., the $k \times k$ submatrix $M[k]$ formed by entries belonging to the intersection of chosen columns and selected rows forms a permutation matrix. Which means that each row in $M[k]$ is a unit vector different from the others. The columns i,k and l in matrix M form 10 vectors of length 3. And while we have unit vectors $(1,0,0)$ in row 2 and $(0,1,0)$ in row 4, none of the rows contains vector $(0,0,1)$. In consequence we cannot construct a permutation matrix on the basis of columns i,j and k. Which means that M is not a $(3,3,n)-$selector.

5.B One of the most important considerations in the message-passing model for distributed computing is the assumption we make about the synchronisation of processors in the network. The most used models in distributed algorithm design are: *synchronous model* and *asynchronous model.*
In *synchronous model* each processor has internal clock and the clocks of all the processors are synchronised. We also assume that the speed of all the processors is the same and that it takes the same amount of time to send a message through any connection in the network.
In *asynchronous model* there is no assumption about the speed of internal clocks or connections. In this model the steps of an algorithm are determined by *conditions* or *events.* We assume that each communication channel (edge) acts as a FIFO queue (a buffer) in which the messages arrive in the same order they are sent. In this model we use *fairness* assumption that guarantees that if a processor P has an event enabling P to perform a task, then P will eventually perform that task.

5.C The periodicity lemma states that if any string $w = w[0..n-1]$ has two periods p and q, s.t., $p+q \leq n$, then string w has also a period $gcd(p,q)$, where $gcd(a,b)$ stands for the greatest common divisor of two positive integers a, b.