PhD Qualifier Examination

Department of Computer Science and Engineering

Date: 21-Mar-2018 **Maximum Marks:** 100

[*Answer any five questions from Group A, and any five questions from Groups B and C.*]

Group A

A.1 Write only the answers to the following parts. No justifications are needed. **(2 × 5)**

(a) Let us take logarithms to the base 2. Define $\log^{(k)} n = \underbrace{\log(\log(\cdots(\log(n))\cdots))}_{k \text{ times}}$, and $\log^* n = \min(k \mid \log^{(k)} n \leqslant 1)$. Find the largest integer n, for which $\log^* n = 4$.

(b) The running time of an algorithm satisfies the recurrence $T(n) = 3T(n/3) + cn$ for some constant $c > 0$. What is $T(n)$ in the big-Oh notation?

(c) An array A consists of n integers each in the range $1 \ldots k$. Your task is to find out whether A contains duplicate elements. What is the worst-case running time (in the big-Oh notation) of the best possible algorithm for solving this problem in the following two cases?

(i) $k = n$ (ii) $k < n$

(d) Let T be a binary tree, and v any node in T. If v has a left child, then $key(leftchild(v)) < key(v)$, and if v has a right child, then $key(v) < key(rightchild(v))$. Demonstrate by an example that T need not be a binary search tree.

(e) You are given an undirected graph $G = (V, E)$ with $|V| = n$ in the adjacency-matrix representation. What is the worst-case running time of depth-first search in G?

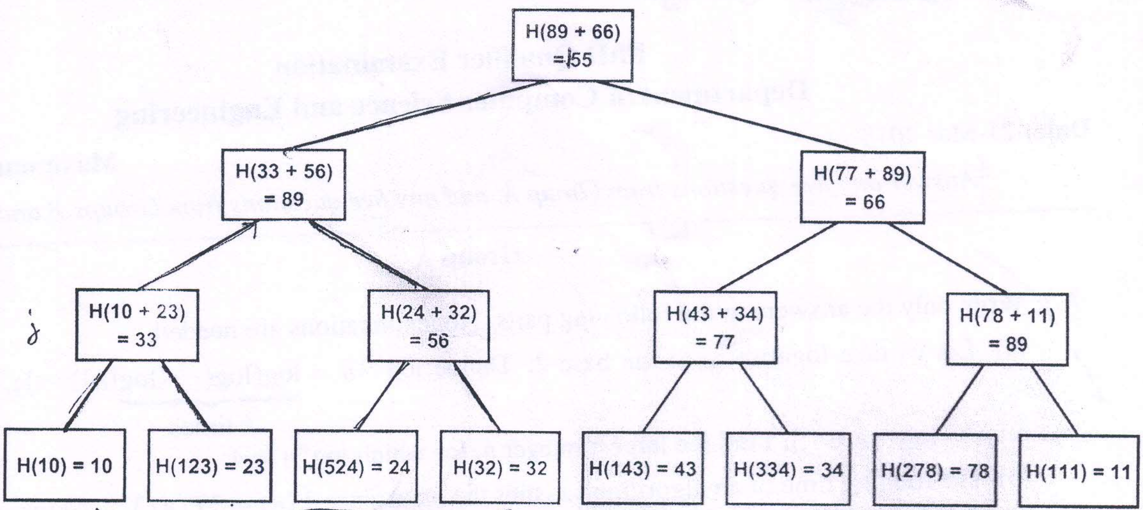
A.2 You are given a sequence $a_0, a_1, a_2, \ldots, a_k$ of distinct integers. It is claimed that this is the sequence of keys encountered while searching for a_k in a binary search tree (BST). Propose an efficient algorithm to verify whether the claim can be justified or not, that is, whether it is possible to encounter the given sequence of keys while searching for the last element of the sequence in a BST. For example, the sequences 3, 8, 9 and 3, 8, 5 are possible, whereas the sequences 3, 1, 7 and 3, 8, 1 are not possible. What is the running time of your algorithm? **(8+2)**

A.3 Let $A = (a_1, a_2, \ldots, a_n)$ be an unsorted array of n integer values. Consider the prefix sums $s_i = a_1 + a_2 + \cdots + a_i$ for $i = 1, 2, 3, \ldots, n$. Your task to sort the array $S = (s_1, s_2, \ldots, s_n)$. If all a_i are positive (or non-negative), then S is already sorted. But it is given that A contains a constant number of negative elements. Propose an algorithm that, given A, outputs the sorted version of S in $O(n)$ time. **(10)**

A.4 You have a box of n chocolates. On each day, you eat one, two, or three chocolates. You want to finish all the chocolates in exactly k days. You want to determine in how many ways this can be done. For example, for $n = 6$ and $k = 3$, all the possibilities are $1+2+3$, $1+3+2$, $2+1+3$, $2+2+2$, $2+3+1$, $3+1+2$, and $3+2+1$, so the desired count is 7. Propose an efficient algorithm to solve this problem. Deduce the running time of your algorithm. **(8+2)**

A.5 Each edge of an undirected graph G is given one of the two colors: Red and Blue. A cycle in G is called monochromatic if all the edges on the cycle have the same color. Propose an efficient algorithm to determine whether G contains a monochromatic cycle. What is the running time of your algorithm? **(8+2)**

A.6 A *Merkle tree* is a tree where the leaf nodes are the hash of the data points, and the non-leaf nodes are the hash of the combined values of its children. Consider a hash function $H(D) = D \% 100$, where D is an integer data point, and % is the remainder after division (the modulo function). Assume that this hash function is used to construct a *binary Merkle tree*. This is a full binary tree of height h storing 2^h data points. A leaf node of the tree stores $H(D)$, where D is an integer data point, and a non-leaf node stores $H(L+R)$, where L and R are the values stored in its left child and right child, respectively. A binary Merkle tree of height three with data points 10, 123, 524, 32, 143, 334, 278, 111 and with $H(D)$ defined as above is given on the next page. Notice that the data points need not be sorted.



(a) Write a user-defined C data type to store a node in a binary Merkle tree. **(2)**

(b) Write a C function that, given a positive integer h and an array A of 2^h distinct positive integer values, creates a binary Merkle tree of height h. The data points (their hash values actually) are stored from left to right in the last level in the same order as they appear in the input array A. Use the hash function $H(D) = D \% 100$. **(8)**

A.7 You are given a two-dimensional matrix of size $n \times n$, where all the elements are distinct integers between 1 and n^2, and every integer appears exactly once. Starting from any cell in this matrix, you can move to left, right, up, or down in a non-wrap-around fashion and without leaving the matrix. A path in the matrix is a sequence of cells based on the above four moves. Starting from any cell, a maximum-length path is defined as the largest sequence of cells such that the values in the cells along the path are in increasing order with an increment by one encountered in each move. As an example, consider the following 3×3 matrix.

	Column 0	Column 1	Column 2
Row 0	2	5	1
Row 1	3	6	9
Row 2	4	7	8

Starting from the cell (Row 0, Column 1), the longest path is 5-6-7-8-9 (Moves: Down-Down-Right-Up). Write a recursive C function to find out the number of cells (not the number of moves) in the longest path starting from a given cell of a given matrix. For example, for the above 3×3 matrix with the input cell $(0, 1)$, the output is 5. **(10)**

A.8 A sentence is called *word-wise palindromic* if the sentence reads the same word-wise both forward and backward. For example, the sentence "*king are you glad you are king*" is word-wise palindromic, whereas the sentence "*was it a car or a cat i saw*" is not word-wise palindromic, although it is character-wise palindromic (ignoring the spaces). Write a C function to check whether a given sentence, comprising lower-case letters and spaces only, is word-wise palindromic. Assume that in the input string, each two consecutive words are separated by a single space, that is, the input does not contain two or more consecutive spaces. **(10)**

Group B

B.1 Define a relation R on $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$ by $(x, y) \in R$ if $x - y$ is a multiple of 5.

(a) What is $|R|$, that is, how many pairs are there in R? **(3)**

(b) Show that R is an equivalence relation on A. **(3)**

(c) Determine the equivalence classes and the partition of A induced by R. **(4)**

B.2 Let C_n denote the number of strings over $\{0, 1, 2\}$ of length n, that contain even numbers of 1's.

(a) Write a recurrence relation and the initial condition(s) that define the sequence $C_0, C_1, C_2, \ldots$. **(5)**

(b) Solve the recurrence relation of Part (a) to obtain an explicit formula for C_n. **(5)**

B.3 (a) The probabilities of winning an assembly election for three candidates A, B, C are 0.6, 0.1, and 0.3, respectively. If A wins, the probability that a new hospital will be built up in the constituency is 0.7, and the corresponding probabilities for B and C are 0.4 and 0.5. What will be the probability that the hospital will be built up? **(5)**

(b) There are two groups of subjects, one of which consists of 5 Science subjects and 3 Engineering subjects, and the other consists of 3 science subjects and 5 Engineering subjects. An unbiased dice is rolled. If it shows a number less than or equal to 4, a subject is selected at random from the first group, but if it shows a number more than 4, the subject is selected at random from the second group. If an Engineering subject is selected, find the probability that it has been done from the first group. **(5)**

B.4 Prove or disprove whether the following languages are regular.

(a) $L_1 = \{w \mid w \in \{0,1\}^* \text{ is not a palindrome}\}$ **(5)**

(b) $L_2 = \{w \mid w \in \{0,1\}^* \text{ contains equal number of occurrences of the substrings 10 and 01}\}$ **(5)**

B.5 Provide the following constructions.

(a) Construct a DFA for the set of all strings over the alphabet $\{0, 1\}$ that, when interpreted in reverse as binary integers, are divisible by 5. Examples of strings in this language are 0, 10011, 1001100, 101, where strings like 111 and 11001 are not in the language. Note that $10011 = 1+8+16 = 25$ and $11001 = 1+2+16 = 19$ if the strings are interpreted in reverse as binary integers. **(5)**

(b) Provide a context-free grammar for $L_3 = \{0^i 1^j 2^k \mid i, j, k \geqslant 0, \text{ and } k \leqslant i \text{ or } k \leqslant j\}$. **(5)**

B.6 Answer the following questions.

(a) Construct a push-down automaton for the language $L_4 = \{a^n b^m \mid n, m \geqslant 0 \text{ and } m < 2n < 3m\}$. **(6)**

(b) Let L_5 be the language of the regular expression $a^*b^* + b^*a^*$. Is $L_5 = \{a, b\}^*$? If not, give an example of a string from $\{a, b\}^*$ that is not in L_5. Design an NFA with four states to accept L_5. **(4)**

Group C

C.1 (a) Minimize the following 4-variable switching function using Karnaugh map, and hence realize it using NAND gates only.

$$F(a,b,c,d) = \sum m(0, 2, 5, 8, 14) + \sum d(3, 4, 10, 15)$$ **(4)**

(b) Implement the function $G(a,b,c) = a'bc + b'c' + ab'$ using 2-to-1 multiplexers only. **(3)**

(c) Draw the gate-level schematic of an SR flip-flop, and clearly explain what is meant by race condition in such a flip-flop. **(3)**

C.2 (a) Design a 4-bit synchronous counter to count in the following sequence: 0000, 0011, 0101, 1001, 0110, 1010, 1100, and then again back to 0000. If the count value is anything other than the above, the counter will reset to 0000 at the next clock. Use JK flip-flops for implementing the states. **(5)**

(b) You are given modulo-10 counter modules with asynchronous set and clear inputs, and running with a synchronous clock. You are required to implement a modulo-60 counter. Show the schematic diagram to achieve this. **(3)**

(c) With respect to switching functions, define the terms: (i) unate function, and (ii) prime implicant. Give examples. **(2)**

C.3 (a) Write expressions for carry generate and carry propagate functions of a full adder. Hence design a 4-bit carry-lookahead adder. Show the steps of the design, and the final gate-level schematic diagram. **(5)**

(b) Represent $+25$ and -15 using 6-bit 2's complement representation. Hence show the steps of multiplying these two numbers using Booth's algorithm. **(3)**

(c) List some distinguishing features of a Reduced Instruction Set Computer (RISC) architecture. **(2)**

C.4 (a) Distinguish between the terms hardware interrupt, software interrupt, and exception. **(3)**

(b) Consider a 2-level memory hierarchy consisting of cache memory and main memory. Derive an expression for the effective memory access time. Clearly state the assumptions you make. **(3)**

(c) A computer system has a word addressable main memory consisting of 4M 32-bit words. It also has a 4K-word cache organized in set-associative manner, with 8 blocks per set and 32 words per block. How many address lines are there in the processor, and how are they partitioned into TAG, SET and WORD fields. Make relevant assumptions. **(4)**

C.5 (a) Define a semaphore. **(3)**

(b) A shared variable x can be accessed by a group of reader and writer processes. A reader process only reads the variable, whereas a writer process only writes the variable. Multiple reader processes should be allowed to read the variable at the same time concurrently; however, if a writer process writes to the variable, no other reader or writer process should be allowed to access the variable until the write finishes. Also, writer processes should be given preference over reader processes, meaning that if both reader and writer processes are waiting to access the variable, the writer processes should be allowed to write before any reader process is allowed to read (however, if some reader processes are already reading when a new writer process arrives, the reader processes will be allowed to finish). Write the code for the reader and the writer processes using semaphores for synchronization. **(7)**

C.6 Consider a demand-paged memory-management system with one-level paging. With respect to this system, answer the following questions briefly.

(a) Where are the page tables of the processes stored (one sentence only)? **(1)**

(b) Name four bits that are kept with each page-table entry for a demand-paged memory-management system and their functions (one sentence only for each bit). **(2)**

(c) How does the operating system switch from the page table of a Process A to that of a Process B, when a context switch changes the running process from A to B? **(2)**

(d) What is a TLB and what is it used for? **(2)**

(e) How can two processes A and B share a page (say, for implementing a shared variable)? **(3)**
