PhD Qualifier Examination

Department of Computer Science and Engineering

Date: 30-Oct-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.

(a) Consider the functions $f(n) = n2^n$ and $g(n) = 3^n$ of non-negative integers n. Which of the following statements is true? **(2)**

(i) $f(n)$ is $O(g(n))$ and $g(n)$ is $O(f(n))$.
(ii) $f(n)$ is $O(g(n))$ but $g(n)$ is not $O(f(n))$.
(iii) $g(n)$ is $O(f(n))$ but $f(n)$ is not $O(g(n))$.

(b) The running time of an algorithm satisfies the recurrence $T(n) = T(n/2) + c\log n$ for some constant $c > 0$. What is $T(n)$ in the big-Θ notation? **(2)**

(c) Let T be a binary with the nodes storing different integer-valued keys (negative keys are allowed). For a node u of T, let $\text{sum}(u)$ denote the sum of all the keys stored in the subtree rooted at u. It is given that $\text{sum}(\text{leftchild}(v)) < \text{key}(v) < \text{sum}(\text{rightchild}(v))$ at every node v of T. Demonstrate by an example that T need not be a binary search tree. **(3)**

(d) Let $c(n)$ denote the smallest positive integer such that any undirected graph with n vertices and $c(n)$ edges must be connected. Write the formula for $c(n)$ as a function of n. **(3)**

A.2 Let A be a two-dimensional $n \times n$ array of integers. Suppose that each row/column of A is sorted (in the ascending order from lower to higher indices). You are given a search key x. Your task is to determine whether x is stored in A. Propose an $O(n)$-time algorithm to solve this problem. **(10)**

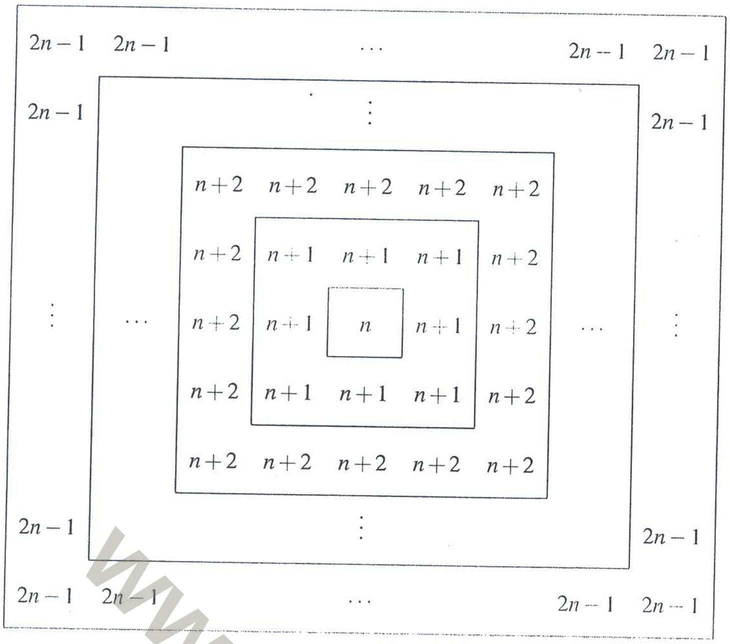
A.3 Let T be a binary search tree (BST), in which every node stores the height of the subtree rooted at that node. Assume that the nodes in T do not maintain parent pointers. Modify the standard BST-insertion algorithm so that after each insertion, the height information at all the nodes of T are updated correctly. Your modified insertion algorithm should take the same running time (in the big-O notation) as the standard BST-insertion algorithm. **(10)**

A.4 Shetti-ji takes orders of n items from customers. He gets the buying prices $b_1, b_2, \ldots, b_n$ of these items from the distributor. He promises selling prices $s_1, s_2, \ldots, s_n$ of the items to his customers. Assume that all s_i, b_i are positive integers, and that $s_i > b_i$ for all $i = 1, 2, \ldots, n$. Shetti-ji buys and sells the items one by one. He has an initial capital of C. As he buys and sells item i, the profit $s_i - b_i$ is added to his current capital. At any point of time, he can buy an item only if its buying price is no more than his current capital. Some items may be so costly that Shetti-ji has to say regret to the customers who ordered those items. Propose an efficient algorithm for Shetti-ji to maximize the number of items that he can sell. Justify the correctness of your algorithm. Work out the running time of your algorithm. **(5+3+2)**

A.5 Professor Nag has written all the chapters for his latest book. If Chapter j requires the results of Chapter i, we say that Chapter i is a prerequisite for Chapter j, and the book should put Chapter i before Chapter j. Professor Nag works out all these prerequisite constraints. Now, his task is to reorganize the chapters so that all the prerequisite constraints are satisfied. Propose an efficient algorithm to help Professor Nag find a desired rearrangement of the chapters or to tell him that no reorganization can meet all the prerequisite constraints. If c is the number of chapters, and p the number of prerequisite constraints, what is the running time of your algorithm (in terms of c and p)? **(8+2)**

A.6 Consider a numeric ring-grid with the numbers from n to $2n-1$ (n is an integer greater than 1). The numbers are placed in a $(2n-1) \times (2n-1)$ grid pattern, following a ring structure with different

Figure 1: Illustrations for A.6



(a) The grid pattern

9	9	9	9	9	9	9	9	9
9	8	8	8	8	8	8	8	9
9	8	7	7	7	7	7	8	9
9	8	7	6	6	6	7	8	9
9	8	7	6	5	6	7	8	9
9	8	7	6	6	6	7	8	9
9	8	7	7	7	7	7	8	9
9	8	8	8	8	8	8	8	9
9	9	9	9	9	9	9	9	9

(b) $n = 5$

numbers. The center of the ring-grid contains the single number n. This number is surrounded by the number $n+1$ forming a 3×3 grid pattern. This ring is then surrounded by another ring with the number $n+2$ forming a 5×5 grid structure, and so on. This continues up to the outermost ring composed of the number $2n-1$ in a $(2n-1) \times (2n-1)$ grid pattern. Part (a) of Figure 1 illustrates this structure. Part (b) of the figure shows the ring-grid for $n = 5$. Write a C program to generate the ring-grid pattern for an input of $n \geqslant 2$ from the user. **(10)**

A.7 Write a recursive C function to merge two alphabetically sorted strings, such that the merged string is also sorted alphabetically. For example, if the two strings are "ALMOST" and "ACCESS" (note that these two strings are alphabetically sorted), then the merged string should be "AACCELMOSSST" (the output is also alphabetically sorted). You can assume that the following function is given to find out the length (number of characters in the string excluding the null character) of a given string.

```
int strlen(char* str);
```

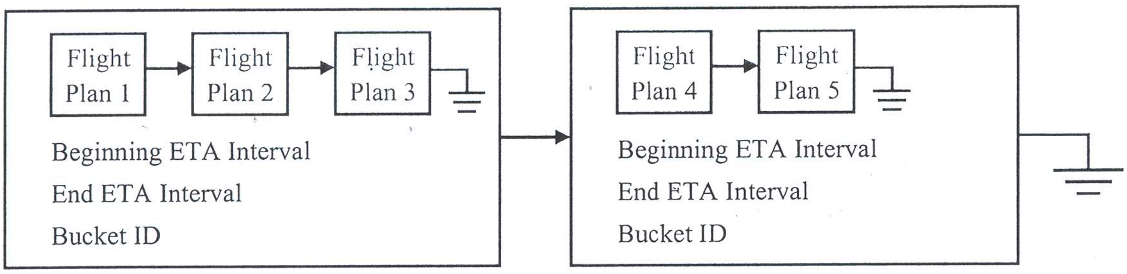
(10)

A.8 Consider an in-air flight-management platform of an air-traffic control (ATC) system, which maintains a digital dashboard for the in-air flights (the flights that have been departed but not yet arrived). The digital dashboard contains the list of in-air flights bucketed in 60 minutes intervals based on the expected time of arrival (ETA) of the flights. Let the current time be 2:00 am. Then, the flights whose ETAs are in between 2:00 am – 2:59 am are kept in the first bucket, the flights whose ETAs are in between 3:00 am â 3:59 am are kept in the second bucket, and so on. Some buckets may be empty if there is no flight to arrive in that time interval. The system maintains a linked list of *active* buckets (the buckets which have at least one flight) sorted by the ETA intervals. Every bucket maintains the bucket ID (an integer), the beginning of the ETA interval, the end of the ETA interval, and a list of flight-plans. A flight-plan contains the flight ID (an integer), the actual departure time, and the ETA. In a bucket, the flight-plans are maintained as a linked list sorted by the actual departure time of the flight. A sample implementation of the dashboard is given in Figure 2.

(a) Define a C structure to represent a bucket for the digital dashboard. Assume that all the times (ETA, departure time) are maintained using a C structure type `TIME`. **(3)**

(b) Consider that a digital dashboard is given (a linked list of buckets). Write a C function to insert a new flight-plan in the digital dashboard. Recall that inside a single bucket, the flight-plans are sorted

Figure 2: Illustration for A.8



by the actual departure time. On the other hand, the buckets are sorted by the ETA intervals. Assume that the new flight-plan belongs to an active bucket. Suppose also that the following two C functions are available: (i) `int timediff(TIME A, TIME B)` which returns the time difference $A - B$ between `TIME A` and `TIME B` in minutes, and (ii) `int maxtime(TIME A, TIME B)` which returns 0, 1 or -1 according as both the times are same, `A` precedes `B`, and `A` succeeds `B`, respectively. **(7)**

Group B

B.1 Let R be the relation defined on the set of eight-bit strings by $s_1 \, R \, s_2$ if and only if s_1 and s_2 have the same number of zeros.

(a) Show that R is an equivalence relation. **(4)**

(b) How many equivalence classes are there? **(3)**

(c) List one member of each equivalence class. **(3)**

B.2 (a) Let $L_0, L_1, L_2, L_3, \ldots$ denote the Lucas numbers, where

$$\begin{aligned} L_0 &= 2, \\ L_1 &= 1, \\ L_{n+2} &= L_{n+1} + L_n \quad \text{for } n \geqslant 0. \end{aligned}$$

(i) Write the first eight Lucas numbers. **(2)**

(ii) For all $n \geqslant 1$, prove that $L_1^2 + L_2^2 + L_3^2 + \cdots + L_n^2 = L_n L_{n+1} - 2$. **(4)**

(b) Prove that of any five points chosen within an equilateral triangle of side length 1, there are two whose distance is at most $\frac{1}{2}$. **(4)**

B.3 (a) Assume that the elephant population of India is 200 at time $t = 0$ and 220 at time $t = 1$, and that the increase from time $t - 1$ to time t is twice the increase from time $t - 2$ to time $t - 1$. Write a recurrence relation and the initial conditions that define the elephant population at time t, and then show that the elephant population at time t is exponential in t. **(5)**

(b) A man alternatively tosses a coin and throws a dice, beginning with the coin. Find the probability that he will get a head before he gets a five or six on the dice. **(5)**

B.4 Prove or disprove the following.

(a) $L_1 = \{a^i b^j c^k \mid i, j, k \geqslant 0, \text{ and if } i = 1, \text{ then } j = k\}$ is a regular language. **(6)**

(b) Let L be a non-regular language. Then $L_2 = L^* = \{w^i \mid w \in L, \text{ and } i \geqslant 0\}$ is again a non-regular language. **(4)**

B.5 Provide the following constructions.

(a) Construct a regular expression for the set of all binary strings such that the number of 0's in the string is divisible by 5. **(4)**

(b) Construct a PDA for the language $L_3 = \{a^i(bc)^j \mid i,j \geqslant 0 \text{ and } i \geqslant j\}$. **(6)**

B.6 (a) Give an NFA with four states equivalent to the regular expression $(01 + 010111)^*$. Convert this automaton to an equivalent DFA. **(5)**

(b) Provide a context free grammar for $L_4 = \{a^n b^m \mid n \neq m-1\}$. **(5)**

Group C

C.1 (a) Find the minimal sum-of-products expression for the following function using the Karnaugh-map method.

$$F(w,x,y,z) = \sum(1,2,3,5,13) + \sum_{\phi}(6,7,8,9,11,15).$$

Here, $\sum_{\phi}$ denotes the don't-care minterms. **(4)**

(b) Show a gate-level implementation of the minimal sum-of-products expression as obtained in Part (a), using NAND gates only. **(2)**

(c) Realize the following function using an 8-to-1 multiplexer, and an additional gate, if required. **(4)**

$$G(w,x,y,z) = wxy' + x'z + w'z'.$$

C.2 (a) Show how you can convert **(4)**

(i) a J-K flip-flop to a T flip-flop, and
(ii) a D flip-flop to an S-R flip-flop.

(b) Three binary counters, modulo-20, modulo-10 and modulo-5 respectively, are connected in cascade. The combination counts modulo-N. What is the value of N? **(2)**

(c) What are the maximum and minimum numbers that can be represented in: **(2)**

(i) n-bit 1's-complement representation, and
(ii) n-bit 2's-complement representation?

(d) How do you convert an 8-bit 2's-complement number to 32-bit 2's-complement representation? **(2)**

C.3 (a) Show the schematic diagram for implementing a 4-bit ripple-carry adder. Assuming that the delay of all the basic gates (AND, OR, NOT) is unity, estimate the worst-case delay of the 4-bit adder. **(5)**

(b) Explain in a step-wise manner what happens when an interrupt signal is applied to a processor. **(3)**

(c) Justify with reasons which mode of I/O is faster: interrupt-driven or DMA. **(2)**

C.4 (a) What do you mean by locality of reference? What is the difference between temporal and spatial locality? **(3)**

(b) Explain the (i) direct and the (ii) set-associative cache-block mapping algorithms. Hence state how the address lines are partitioned into various logical fields (like TAG, SET, WORD, and so on). **(3)**

(c) A computer system uses 16-bit memory addresses. It has a 2K-byte cache organized in a direct-mapped manner with 64 bytes per cache block. Assume that the size of each memory word is 1 byte. When a program is executed, the processor reads data sequentially from the following word addresses:

128, 144, 2176, 2180, 128, 2176.

All the above addresses are shown in decimal values. Assume that the cache is initially empty. For each of the above addresses, indicate whether the cache access will result in a hit or a miss. **(4)**

C.5 (a) Consider the following set of jobs with given arrival times and duration.

Job Id	Arrival Time	Duration
1	0	10
2	1	7
3	2	5
4	2	4
5	3	5
6	4	10

Show the Gantt chart if the jobs are scheduled on a single CPU using the Shortest Remaining Time First (same as preemptive Shortest Job First) scheduling policy. Also calculate the average waiting time of the jobs. **(5)**

(b) Let a Process *A* create a shared integer *x*, initialize it to zero, and then fork ten child processes. Each child process simply increments the variable *x*. The parent process should wait for all ten child processes to finish, and then prints the shared integer *x*. All the child processes run an identical code. Write the code for Process *A* (including creating child processes) and for any one child process to implement the above synchronization. You can only use (i) binary semaphores and the associated wait/signal system calls, and (ii) other shared variables (just identify them as shared in your program). You are not allowed to use counting semaphores or any special system calls like `wait()`. **(5)**

C.6 (a) What is a Translation Lookaside Buffer (TLB)? **(2)**

(b) Clearly explain step by step how a virtual address will be translated to a physical address in a demand-paged memory-management system with 32-bit virtual address and 4 KB page size. Your answer should have sufficient details to outline how each step can be implemented. **(5)**

(c) Consider a system with no logical addresses (that is, all memory addresses used are direct physical addresses). Suggest any scheme which can prevent one process from overwriting the code/data of another process in memory. **(3)**

— END —
