PhD Comprehensive Examination
Department of Computer Science and Engineering

Date: 07-May-2012 **Time:** 4 Hours **Maximum Marks:** 100

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

Group A

A.1 (a) Write the asymptotic upper and lower bounds of $T(n) = T(9n/10) + \Theta(n)$. No explanation is needed.

(b) Write a C function `int allsq(int` n`)` that returns the square numbers $1, 4, 9, \ldots, n^2$. The function should only use addition (no multiplication).

(c) Draw a **binary search tree** T whose pre-order traversal produces $\langle 9, 4, 6, 20, 17, 11 \rangle$.
Find the average number of comparisons for insertion of the 7th node in T.

(2 + 3 + 5)

A.2 (a) Define an adjacency matrix A for a connected and undirected graph $G(V, E)$.

(b) Suggest some algorithm to prepare a binary matrix B from A such that $B[i][j] = 1$ if and only if there exists some vertex $k \in V - \{i, j\}$ such that $(i, k) \in E$ and $(k, j) \in E$.
Explain its time complexity.

(2 + 8)

A.3 Suggest an algorithm to determine whether two unsorted arrays A and B, of size n_1 and n_2 respectively, are disjoint. You should try to reduce the time complexity of your algorithm. Explain its time complexity. **(8+2)**

A.4 Consider an existing linked list where each node contains an integer and a pointer to the next element of the linked list. The linked list is *self–adjusting*, in a sense that whenever an integer value is searched for in the linked list and the search succeeds, the node containing the value is moved to the front of the linked list, without changing the relative order of the other elements in the linked list. Write a C function `void SearchAndAdjust (node *head, int val)` to search for a value val in the linked list, and if found, to make the necessary adjustments to retain the *self–adjusting* property. If the value is not found, the linked list is left untouched. Here $head$ is a pointer to the current beginning node of the linked list. **(10)**

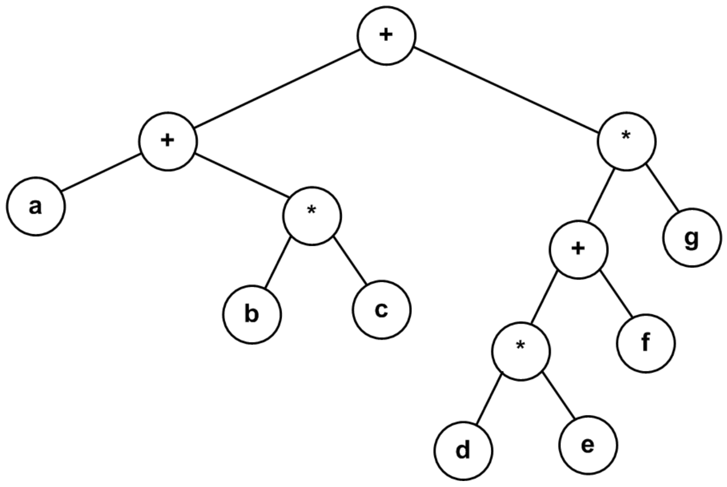
A.5 Consider a game where N persons numbered 0 to $N - 1$ are sitting in a circle. Starting at person 0, a hot potato is passed to the next person at each step (in the direction of increasing person number, modulo N). After M passes, the person holding the hot potato is eliminated, the circle closes rank, and the game continues with the person sitting after the eliminated person picking up the hot potato. The last remaining person wins. Thus, if $M = 1$ and $N = 5$, the order of elimination is 1, 3, 0, 4, and the winner is 2. You are required to solve this problem using circular linked list(s).

(a) Declare a C structure for each node of the linked list called `person` that contains: an integer containing the person number, and a pointer to the next node representing the person who is to be passed the hot potato next.

(b) Write a C function to return the winner number. The function prototype is `int winner (person *head, int M, int N)`, where `head` is a pointer to person–1 at the beginning of the game. Make sure you free the corresponding memory whenever a person is eliminated.

(2 + 8)

A.6 Consider an “’expression tree” to evaluate integer arithmetic expressions. Each node of such a binary tree contains either one of the commutative arithmetic binary operators “*” and “+”, or an integer. For example, the expression “$a \star b$” is represented by a binary tree with “$\star$” at the root node and a and b at the two child nodes. Write a C function `int evaluate (node *root)` to evaluate and return the value of a given expression tree, where $root$ is a pointer to the root node of the tree. The function recursively evaluates the left and right sub–trees, and then combines the values using the operator at the *root* node, to evaluate the entire expression. An example expression tree for the expression $(a + b \star c) + ((d \star e + f) \star g)$ is shown in Fig. . **(10)**



Group B

B.1 Let T_n be a sequence of positive integers defined recursively as:

$$T_0 = 2$$
$$T_n = T_{n-1}^2 + T_{n-2}^2 + \ldots + T_1^2 + T_0^2 \text{ for all } n \geq 1$$

Prove the following assertions. You may use induction on n, whenever necessary.

(a) $T_n = T_{n-1}(T_{n-1} + 1)$ for all $n \geq 2$

(b) $T_n \geq 2^{2^n}$ for all $n \in N_0$

(5 + 5)

B.2 (a) Prove that if m and n are relatively prime and mn is a perfect square, then m and n are each perfect squares.

(b) Show that in any set X of people there are two members of X who have the same number of friends in X. (It is assumed that $|X|$ is at least 2, and if x is a friend of y then y is a friend of x)

(5+5)

B.3 (a) Let $f : A \rightarrow B$ be a function and σ an equivalence relation on B. Define a relation ρ on A as: $a \,\rho\, a'$ if and only if $f(a) \,\sigma\, f(a')$. Prove that ρ is an equivalence relation on A.

(b) A binary communication channel carries data as 0 or 1. However, due to noise etc., sometimes a 1 transmited may be received as 0 and vice-versa. Suppose that the probability that a 0 transmitted is received as a 0 is 0.94, and the probability that a 1 transmitted is received as a 1 is 0.91. Also, the probability of transmitting a 0 is 0.45. What is the probability that a 1 was transmitted given that a 1 was received?

(5 + 5)

B.4 Let L_4 denote the set of all strings w over the alphabet $\{a, b\}$ such that w contains at least one a in the last four positions. Design a DFA (deterministic finite automaton) with only five states to accept L_4. **(10)**

B.5 Let L_5 denote the set of all strings over the singleton alphabet $\{a\}$, that are of the form a^n for some composite integer n. Prove that L_5 is not regular. (**Hint:** It may be easier to prove that the complement of L_5 is not regular.) **(10)**

B.6 Let L_6 denote the set of all strings over the alphabet $\{a, b, c\}$, that are not of the form $a^n b^n c^n$ for some $n \geq 0$. Prove that L_6 is a context-free language. (**Hint:** Carefully characterize all strings in L_6.) **(10)**

Group C

C.1 (a) List step by step what happens when a running process executes a ”scanf(”%c”, &ch)” command in a C program, and afterwards when the user presses a key.

(b) Consider a 1-level paging system with a TLB access time of 10 nanonseconds and memory access time of 50 nanoseconds. The page fault handling time is 1 milliseconds. If the TLB hit rate is 0.99, what can be the maximum page fault rate such that the effective memory access time is at most 100 nanoseconds?

(5 + 5)

C.2 (a) What is a system call? Explain exactly how a system call switches a process to kernel mode during its execution and how is it switched back to user mode on return from a system call. (1-2 sentence each max)

(b) Suppose two processes want to execute a common code (say that for \bin\ls in Linux) in a demand-paged system. Explain how the two processes can share a single copy of the code in memory. (1-2 sentence max.)

(c) A context switch in a demand-paged system will require the OS to change the page table to be used for address translation, as the page table of the new process to be run will be used for any further address translation. A naive implementation will require the OS to copy the new process's page table into memory, which will increase the context switch time. How is this usually handled so that page table load time during context switch is reduced? (1-2 sentence max.)

(5 + 3 + 2)

C.3 (a) Explain how can you make the semaphore operations *wait* and *signal* atomic on (i) a uniprocessor machine, and (ii) a multiprocessor machine. (1-2 sentence each max.)

(b) Consider the well-known readers-writers problems in which a set of writers write to a data item and a set of readers read it. It is required that each writer has exclusive access to the data item (so when a writer accesses the data item, no other reader or writer can access it), but multiple readers can access it simultaneously if no writers are accessing it. Design a solution using semaphores (write the code for one reader and one writer).

(4 + 6)

C.4 (a) Using a 4:16 decoder design a logic circuit that generates an active-high output whenever a 4-bit binary word contains exactly two 1s. You are allowed to use another *single* type of gate of your choice.

(b) Consider many 4 MB SRAM chips offering data widths of 8 bits. Show the memory organizations for building two 16 MB memory units, one with 8 bit word and the other 32 bit words.

(5 + 5)

C.5 (a) Multiply using the right shift-add algorithm the following two signed-8 bit numbers: $A = 10101010$, $X = 10111101$.

Show each step of the algorithm. Also identify when there is an overflow, and adopt suitable measures to solve it.

(b) When running an integer benchmark on a RISC machine, the average instruction mix was as follows:

Instructions	Average Frequency
Load	26 %
Store	9 %
Arithmetic	14 %
Compare	13 %
Cond. Branch	16 %
Uncond. Branch	1 %
Call/returns	2 %
Shift	4 %
Logical	9 %
Misc.	6 %

The following measurements of average CPI for individual instruction categories were made:

Instruction type	Average CPI (clock cycles)
All ALU instructions	1
Load-store	1.4
Conditional Branches:	
Taken	2.0
Not taken	1.5
Jumps	1.2

Assume that 60 % of the conditional branches are taken and that all instructions in the Misc. category are ALU instructions. What is the CPI of the benchmark on this RISC machine?

(6 + 4)

C.6 (a) Assume that memory is *byte-addressable*, memory addresses are 32 bits wide, a cache line contains 16 bytes, sets contain 2 lines, and the cache size is 4096 lines (64 KB). Show the various parts of the address and identify which portion of the address is used to access the cache.

(b) Consider a *word-addressable* cache memory of size 8 words. The organization of the cache memory is 2-way set-associative and the word line is 2 words. The cache replacement policy is LRU (Least Recently Used).

Consider further a program reading elements of a 4×4 matrix A, stored in row-major order starting at address 0, and ending at 15.

Depict the address mapping by populating Table 1 and show the contents of the cache at each step of the memory access, and the resulting Hit/Miss pattern by populating the Table 2.

(4+6)

Memory Address	Set	Word
0		
⋮	⋮	⋮
15		

Table 1: Memory Mapping for the 2-way set associative cache

Memory Address	Hit/Miss	Set 0				Set 1			
		Block 0		Block 1		Block 0		Block 1	
		word 0	word 1	word 0	word 1	word 0	word 1	word 0	word 1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 2: Content of the Cache during Matrix Read