Sumana

PhD Qualifier Examination

Department of Computer Science and Engineering

Date: 20-Mar-2014 **Maximum Marks:** 100

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

Group A

A.1 Let G be a directed acyclic graph with n vertices. Let s be a source (a vertex with in-degree zero) and t a sink (a vertex of out-degree zero) in G. We want to compute the number of paths from s to t. For $u, v \in G$, let $N(u,v)$ denote the number of paths in G from u to v. So our task is to compute $N(s,t)$. Let $v_1, v_2, \ldots, v_k$ be all the neighbors of s is G. Then, we have $N(s,t) = N(v_1,t) + N(v_2,t) + \cdots + N(v_k,t)$, that is, we compute each $N(v_i,t)$ recursively, and return their sum.

6

(a) Describe the termination condition(s) for the above recursive algorithm. **(3)**

(b) Prove that the worst-case complexity of the above recursive algorithm can be exponential in n. **(7)**

A.2 (a) Describe what ADT (abstract data type) calls (operations with functional specifications) should be defined for implementing stacks. Do not implement the calls. Only mention the call prototypes. For example, here is a call prototype for inserting an element in an ordered list at a given index:

list insert (list, index, element). **(3)**

(b) Use these stack ADT calls to recognize strings with balanced parentheses. Examples of such strings: (), ((())), ()()(), ()(()(()(()()))). Non-examples: ((), (()(())),)()(. **(7)**

A.3 Let $[P_0, P_1, P_2, \ldots, P_{n-1}]$ be an array of n distinct points in the two-dimensional plane. Each point P_i is specified by its to coordinates x_i, y_i. A point P_i in the given array is called *maximum* if for all $j \in \{0, 1, 2, \ldots, n-1\}$ with $j \neq i$, we have $x_j \leqslant x_i$ and $y_j \leqslant y_i$.

(a) Show that not every array of points contains maximum point(s). **(3)**

(b) Propose an efficient algorithm to find a maximum point in a given array of points (if a maximum exists). What is the running time of your algorithm? **(7)**

A.4 (a) Define a max-heap and its contiguous representation in an array. **(4)**

(b) Convert the array $[20, 11, 25, 7, 23, 17, 9]$ to a max-heap. **(3)**

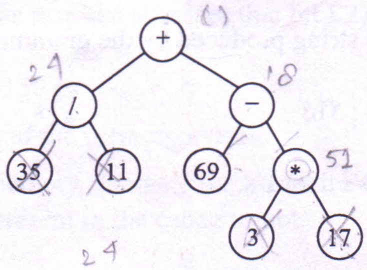
(c) Explain how you insert 22 in the max-heap of Part (b). **(3)**

A.5 Let $\mathbf{u}, \mathbf{v}, \mathbf{w}$ be n-dimensional column vectors (that is, $n \times 1$ matrices), and I the $n \times n$ identity matrix. Propose a $\Theta(n)$-time algorithm to compute the vector $(I + \mathbf{u}\mathbf{v}^{\mathrm{t}})\mathbf{w}$, where $\mathbf{v}^{\mathrm{t}}$ is the transpose of $\mathbf{v}$. Assume that only $\mathbf{u}, \mathbf{v}, \mathbf{w}$ are supplied as input to your algorithm. I is a constant matrix and does not need to be explicitly provided as an input. **(10)**

A.6 An *expression tree* is a binary tree having two properties: (a) every non-leaf node stores a binary operator and has two child nodes, and (b) every leaf node stores a numeric operand. Such a tree stands for an arithmetic expression. For example, the following tree stands for the expression $(35/11) + (69 - (3 * 17))$, and evaluates to 21. Assume that the operands are integers, and that the operators to be supported are $+, -, *, /$ and % with their usual meanings.

Left root

1.2 3 4 5 6



35 ÷ 11

2 child

Every node

(a) What elements you should store in a node of an expression tree? **(3)**

(b) Propose an efficient algorithm to evaluate the expression tree. Report error if the input tree is not a valid expression tree. You do not need to write a C code. A description of your algorithm suffices. **(5 + 2)**

A.7 In a *doubly linked list*, each node is given two pointers, the first pointing to the next element in the list, and the second to the previous element in the list. The next pointer of the last node and the previous pointer of the first node should be NULL.

(a) Describe a suitable C data type to store a node in a doubly linked list. Suppose that each node in the list stores an integer key value. **(3)**

(b) Let L be a *sorted* doubly linked list (of integers). Write a C function to insert an integer x in L. If x already resided in L, then the insertion produces no change. If x was absent from L, it is inserted in the appropriate position so that after the insertion, L continues to remain sorted. **(7)**

A.8 A d-dimensional Gray code is a listing of all of the 2^d binary strings of length d with any two consecutive strings in the list differing in exactly one position. A recursive construction of Gray codes follows. For $d = 1$, it is the listing $\langle 0, 1\rangle$. So suppose $d \geqslant 2$, and let $\langle s_0, s_1, \ldots, s_{2^{d-1}-1}\rangle$ be a $(d-1)$-dimensional Gray code. Then, $\langle 0s_0, 0s_1, \ldots, 0s_{2^{d-1}-1}, 1s_{2^{d-1}-1}, \ldots, 1s_1, 1s_0\rangle$ is a d-dimensional Gray code.

(a) Define a suitable C data type to store an array of 2^d strings each of length d. **(3)**

(b) Write a C function that, upon the input of d, returns the d-dimensional Gray code as a data of type defined in Part (a). **(7)**

Group B

B.1 Determine the number of integers between 1 and 250 that are divisible by (at least) one of the integers 2, 3, 5, and 7. **(10)**

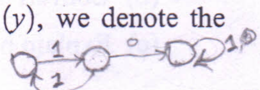
B.2 Find the general solution of $S_n - 4S_{n-1} + 4S_{n-2} = 2^n + n2^n$. **(10)**

$P \quad (1-P)\cdot P \quad (1-P)^2\cdot P$

H, TH, TTH,

B.3 A biased coin (with probability of obtaining a head equal to $p > 0$) is tossed repeatedly and independently until the first head is observed. Compute the probability that the first head appears at an *even* numbered toss. **(10)**

B.4 The following two languages are defined over the alphabet $\Sigma = \{0, 1\}$. By $n_1(y)$, we denote the number of 1's in the string y.



A1

(a) Prove that the language $L_1 = \{1^k y \mid y \in \{0,1\}^*,\ k \geqslant 1,\ n_1(y) \geqslant k\}$ is regular. **(5)**

(b) Prove that the language $L_2 = \{1^k y \mid y \in \{0,1\}^*,\ k \geqslant 1,\ n_1(y) \leqslant k\}$ is not regular. **(5)**

B.5 (a) Construct a DFA comprising at most six states which accepts the language (over $\Sigma = \{0, 1\}$) corresponding to the regular expression $(11 + 110)^*0$. **(6)**

A2

(b) Prove that the language $L = \{0^i 1^j \mid 2i \leqslant j \leqslant 3i\}$ (over $\Sigma = \{0, 1\}$) is context free. **(4)**

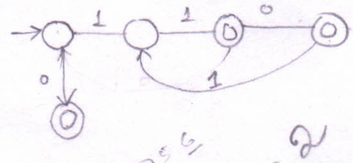
B.6 (a) A grammar $G = (V, T, S, P)$ (all symbols having their usual meanings) is said to be *right-linear* if all productions are either of the form '$A \to xB$' or of the form '$A \to x$'. Prove that if G is right-linear, then $L(G)$ is regular. **(5)**

A3

(b) Inductively prove that every string produced by the grammar

$$S \to 0 \mid S0 \mid 0S \mid 1SS \mid SS1 \mid S1S$$

(with start symbol S) has more 0's than 1's. **(5)**



P1 — 2
P2 — 3
P3 — 5
P4 — x

Group C

C.1 (a) Prove, using Boolean algebraic justifications, that arithmetic overflow is impossible if two n-bit numbers of opposite signs are added in a 2's complement number system. (3)

(b) Consider the Boolean expressions:

$$\begin{aligned} IP &= A + BS_0 + B'S_1, \\ IG &= AB'S_2 + ABS_3, \text{ and} \\ F &= IP \oplus IG \oplus M, \end{aligned}$$

where A, B, M, S_0, S_1, S_2 and S_3 are Boolean variables. Prove, using Boolean algebra, that if $M = 1$,

$$F = A'BS_0' + A'B'S_1' + AB'S_2 + ABS_3. \quad (6)$$

(c) From the above expression for F, explain how F can be used to implement all sixteen two-variable Boolean functions $F(A,B)$. (1)

C.2 We wish to design a sequence detector circuit which detects three or more consecutive 1's in an infinite stream of bits coming through an input line, by making an output line logic-1 (which is otherwise at logic-0).

(a) Draw the state diagram of the finite state machine. (2)

(b) Determine the type of the circuit (Moore or Mealy machine). (1)

(c) Write the complete state transition table of the circuit, assuming minimum number of D flip-flops. Clearly indicate the state encoding you have assumed. (2)

(d) Implement the logic minimized circuit using D flip-flops. (5)

C.3 (a) Identify four distinguishing features of CISC and RISC architectures. (4)

(b) With respect to interrupt-driven I/O, clearly explain the following:

i) If multiple devices are connected to the same interrupt line, how does the CPU identify the source of the interrupt? (2)

ii) Once the interrupting device is identified, how is the corresponding interrupt handler invoked? (2)

iii) How are nested interrupt requests typically handled (arrival of an interrupt request while another one is being processed)? (2)

2^{14} $2^{15} \times 2$

C.4 (a) Four 16K×4 memory modules are to be interconnected to design a 32K×8 memory system. Show a schematic diagram for the same. (3)

(b) In a two-level cache memory system, the parameters are:

Main memory access time: 150 ns
L1 access time: 20 ns
L2 access time: 50 ns
L1 hit ratio: 98.5%
L2 hit ratio: 85.0% (w.r.t. the residual accesses that hit L2)
MM-L2 block transfer time: 500 ns
L2-L1 block transfer time: 150 ns

98.5% 20ns + 2.5% (20 + (85% * 150ns) + 50)
+ (15% (50ns + 500ns))
+ 150

Estimate the average access time of the memory system. (4)

(c) For a set-associative cache memory, estimate the hardware overhead required to identify whether a requested memory location is present in the cache or not. (3)

C.5 (a) Consider three processes (process IDs 0,1,2) with CPU burst times 2, 4, 8 units. All processes arrive at the time zero. Consider the preemptive longest remaining time first (LRTF) scheduling algorithm. In this algorithm, the process with the longest next remaining CPU time will be scheduled first. In LRTF, ties are broken by giving priority to the process with the lower process id. Compute the average turnaround time with the help of Gantt chart. Assume that the time units are integral. **(4)**

(b) Assume that we have a demand paged memory. The page table is held in registers. It takes 8 ms to service page fault if an empty frame is available or if the replaced page is not modified, and 20 ms if the replaced page is modified. Memory access time is 100 ns. Assume that the page to be replaced is modified 70% of the time. What is the maximum acceptable page-fault rate for an effective access time of no more than 200 ns. **(3)**

$200ns = (1-P)\times 100ns + P\times(30\% \times 8ms + 70\% \times 20ms)$

(c) A CPU generates 32-bit virtual addresses. The page size is 4 KB. The processor has a translation look-aside buffer (TLB) which can hold a total of 128 page table entries and is 4-way set associative. What is the minimum size of the TLB tag? **(3)**

$4KB = 2^{12}$

no. of set $\frac{128}{4} = 2^5$

C.6 (a) A shared variable p, initialized to zero, is operated on by four concurrent processes W, X, Y, Z as follows. Each of the processes W and X reads p from memory, increments by one, stores it to memory, and then terminates. Each of the processes Y and Z reads p from memory, decrements by two, stores it to memory, and then terminates. Each process before reading p invokes the *wait()* operation on a counting semaphore S and invokes the *signal()* operation on the semaphore S after storing p to memory. Semaphore S is initialized to two. What is the maximum possible value of p after all processes complete execution? **(4)**

(b) Consider a system which implements (i) preemptive priority-based CPU scheduler and (ii) busy waiting for mutual exclusion. In such a system, explain whether it is possible that a high priority process gets delayed indefinitely because of the presence of lower priority processes? Assume that no process does any I/O and there is no deadlock in the system. **(3)**

(c) A system uses FIFO policy for page replacement. It has four page frames with no pages loaded to begin with. The system first accesses 100 distinct pages in some order and then accesses the same 100 pages but now in the reverse order. How many page faults will occur? **(3)**

$\square\ 2^{12}$

$\frac{p.no}{20}$ $\frac{p.no}{32}$

$32-12$

20 bit for pg no

$128 \times$ size of page table entry