PhD Qualifier Examination
Department of Computer Science and Engineering

Date: 29-Mar-2017 **Maximum Marks:** 100

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

Group A

A.1 Supply short answers to the following parts. **(2 × 5)**

(a) Write the asymptotic tight bound of $T(n)$ defined as

$$T(n) = \begin{cases} T(3n/4)+1 & \text{if } n \geqslant 4, \\ n & \text{otherwise.} \end{cases}$$

(b) A binary search tree contains an odd number of integers. Assume that each node in the tree stores only a key and two child pointers. What is the best-case time complexity to find the median of these integers from the tree?

(c) An unordered one-dimensional array X contains $4n$ distinct elements. We need to partition X into X_1, X_2, X_3, X_4 with n elements in each part, such that they are ordered, that is, $a_1 < a_2 < a_3 < a_4$ for all $a_1 \in X_1, a_2 \in X_2, a_3 \in X_3, a_4 \in X_4$. The elements in the individual parts need not be ordered after the partition. Write the time complexity of the best algorithm for this.

(d) L and M are two linked lists, each containing n distinct elements sorted in increasing order. There may, however, be some elements common to L and M. What is the worst-case time complexity for the construction of a new linked list N that would contain all elements of L and M but without any duplicate?

(e) A star graph of order n is a tree having a central node of degree n and n other nodes of degree one each. Given the adjacency-list representation of a (general) graph G, what would be the worst-case time complexity to decide whether G is a star graph?

A.2 Let $G = (V, E)$ be an undirected graph with n vertices numbered $0, 1, 2, \ldots, n-1$. You maintain a max-priority queue Q of the vertices with the vertex degrees used for heap ordering. Suppose that the graph is dynamic in the sense that new edges are added and existing edges are deleted frequently. Since the addition or deletion of an edge changes vertex degrees, the queue Q should be appropriately and efficiently modified after each such operation. In the following parts, you do not need to consider the storage of G or its modifications necessitated by edge additions and deletions.

(a) What data structures should you maintain in addition to the storage of G? **(3)**

(b) Mention how Q can be efficiently modified after each addition of a new edge. **(5)**

(c) What is the running time of your algorithm of Part (b)? **(2)**

A.3 Prove that any comparison-based sorting algorithm requires $\Omega(n \log n)$ comparisons in the worst case. **(10)**

A.4 Let A be an $m \times n$ matrix of distinct integers, that is, no two entries of the matrix are the same. For each $i \in \{1, 2, \ldots, m\}$, let $f(i)$ denote the index of the column containing the minimum element of row i. It is given that $f(1) \leqslant f(2) \leqslant \cdots \leqslant f(m)$.

Here is a description of a divide-and-conquer algorithm that computes $f(i)$ for each row $i \in \{1, 2, \ldots, m\}$.

"Construct a submatrix A' of A consisting of the even numbered rows of A. Recursively determine the index of the minimum element of each row of A'. Then compute the index of the minimum element in the odd numbered rows in A."

(a) Suppose we know $f(i)$ for each even-numbered row i. Explain how to use this information to compute $f(j)$ for all odd-numbered row j in $O(m+n)$ time. **(5)**

(b) Write the recurrence describing the running time of the divide-and-conquer algorithm described above. Show that the running time of your algorithm is $O(m + n \log m)$. **(5)**

A.5 Let $G = (V, E)$ be an undirected graph on n vertices $v_1, v_2, \ldots, v_n$. Let $N(v_i) := \{v_j \in V \mid (v_i, v_j) \in E\}$ denote the *neighborhood* of v_i, and $N^h(v_i) = \{v_j \in N(v_i) \mid j > i\}$ the *higher neighborhood* of v_i (that is, the set of neighbors with higher indices). Your task is to find out whether for every $i \in \{1, 2, \ldots, n\}$, $N^h(v_i)$ is a clique. (A subset of vertices of a graph is called a *clique* if the subgraph induced on this subset is complete. That is, every two distinct vertices in such a subset are neighbors of each other.) Assume that you are given an adjacency-matrix representation of the graph G.

(a) Design a $\Theta(n^3)$-time algorithm to solve the problem. **(3)**

(b) Design a $\Theta(n^2)$-time algorithm to solve the problem. (**Hint:** Let $N^h(v_i) = S = \{v_j, v_k, v_l, v_m, \ldots\}$ with $i < j < k < l < m < \cdots$. If S is a clique, then $S \setminus \{v_j\} \subseteq N^h(v_j)$, $S \setminus \{v_j, v_k\} \subseteq N^h(v_k)$, and so on.) **(7)**

A.6 Let $A[0 \ldots n-1]$ be an array of n distinct integers. We call A to be of zigzag type if for some indices i, j satisfying $0 < i < j < n-1$, $A[0 \ldots i]$ is sorted in increasing order, $A[i \ldots j]$ is sorted in decreasing order, and $A[j \ldots n-1]$ is sorted in increasing order. For example, the array $[2, 3, 9, 7, 6, 4, 1, 5]$ is of zigzag type with $i = 2$ and $j = 6$. Write a C function that takes A and its size n as input. The function first checks whether A is of zigzag type. If not, it reports *failure*, and returns. Otherwise, it prints the three subarrays $A[0 \ldots i]$, $A[i \ldots j]$, and $A[j \ldots n-1]$ in three separate lines, and returns. **(10)**

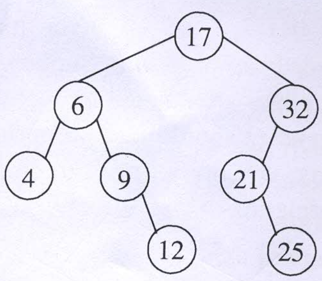
A.7 A sequence $a_0, a_1, a_2, \ldots$ is defined inductively as follows.

$$\begin{aligned} a_0 &= 1, \\ a_{2k+1} &= a_{2k} + a_k + 1 \text{ for all } k \geqslant 0, \\ a_{2k+2} &= a_{2k+1} + a_{k+1} \text{ for all } k \geqslant 0. \end{aligned}$$

(a) Write a recursive C function to compute and return a_n (given n as input). **(5)**

(b) Write an efficient iterative C function to compute and return a_n (given n as input). **(5)**

A.8 An array-based representation of a binary search tree T is demonstrated in the following figure.



Size: 8

Root: 3

	0	1	2	3	4	5	6	7	
Key	9	12	21	17	32	4	25	6	...
Left	−1	−1	−1	7	2	−1	−1	5	...
Right	1	−1	6	4	−1	−1	−1	0	...

We store the size (number of nodes) of T; call it n. We store information about the n nodes in the first n locations of three arrays Key, Left, and Right. Each index $i \in \{0, 1, 2, \ldots, n-1\}$ stores the particulars of one node. Key[i] stores the key stored at this node, Left[i] stores the index of the left child (-1 if this child does not exist), and Right[i] stores the index of the right child (or -1 if no right child exists). The index of the root node is maintained separately. The dotted box in the above figure is such a representation of the tree on the left. The three arrays are not to be kept sorted, that is, any node can appear at any index.

(a) Propose a user-defined data type to store a binary search tree in this format (the dotted box in the above figure). **(3)**

(b) Write an efficient C function to insert a key x into a binary search tree T in this representation. Recall that if x already resides in T, no change is made. Otherwise, a new node is inserted at an appropriate position such that the binary-search-tree ordering is maintained. **(7)**

Group B

B.1 Fibonacci numbers are defined by $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$ for $n \geqslant 2$. Prove that

$$F_0F_1 + F_1F_2 + \ldots + F_{2n-1}F_{2n} = F_{2n}^2$$

for any positive integer n. **(10)**

B.2 An integer is called *square-free* if it is not divisible by a^2 for any integer $a > 1$. How many integers in the range $1, 2, 3, \ldots, 200$ are square-free? **(10)**

B.3 We randomly choose non-negative integers x_1, x_2, x_3, x_4 that satisfy the equation $x_1 + x_2 + x_3 + x_4 = 8$. We assume that each solution has an equal probability of being chosen. Given that at least one of x_1 and x_2 is equal to 1, what is the probability that $x_1 = 1$? **(10)**

B.4 Answer the following questions.

(a) Consider the language $L_1 = \{a^i b^j c^k \mid i \neq j \text{ or } i \neq k\}$ over $\Sigma = \{a, b, c\}$. Define formally $\overline{L_1}$, the complement of this language. You should not use trivial definitions like $\overline{L_1} = \{x \in \{a, b, c\}^* \mid x \notin L_1\}$. **(5)**

(b) Prove or disprove that $\overline{L_1}$ is a regular language. **(5)**

B.5 Provide the following constructions.

(a) A DFA for accepting strings over $\{0, 1\}^*$ such that each string contains either odd number of 0's or odd number of 1's. **(4)**

(b) A CFG for the language $L_2 = \{a^i b^j c^k \mid j \neq i + k\}$. **(6)**

B.6 Prove or disprove the following statements.

(a) $L_3 = \{w \in \{a, b\}^* \mid w \text{ is of odd length and starting symbol of } w \text{ is identical with its middle symbol}\}$ is regular. **(6)**

(b) Let L_4 be the language of all palindromes over $\Sigma = \{a, b\}$, which is known to be non-regular. Then, "Any infinite subset L_5 of L_4 such that every string in L_5 contains at least one a and one b is also non-regular." **(4)**

Group C

110 011 101

C.1 (a) A three-input gate G realizes a function $f(A, B, C) = ABC' + A'BC + AB'C$. Show that the gate G along with the logic value 1 are functionally complete. **(3)**

(b) Write the SOP (Sum-of-Product) expression of the complement of the output of G. **(2)**

(c) Realize the following function by means of two G gates:

$$Z(A, B, C, D) = \Sigma(0, 1, 2, 4, 7, 8, 9, 10, 12, 15).$$ **(5)**

C.2 We want to design a counter using only D flip-flops, that goes through the states $0, 1, 2, 4, 0, \ldots$. If a value other than $0, 1, 2, 4$ appears at any time, the next state should be 0 in the next clock pulse.

(a) Draw the state diagram for the counter. **(2)**

(b) Show the excitation table, and write the expressions for excitations. **(4)**

(c) Draw the circuit diagram of the counter. **(4)**

C.3 A specialized monitoring program needs to process an unbounded stream of incoming data (as 8-bit 2's complement integers) from an input peripheral device. The data is to be stored in a circular buffer of size N bytes starting at address `BBase`. At any instant, the index of the start of the data stream in the circular buffer is to be maintained at location `BBase` $+N$ and that of the end at `BBase` $+N+1$ (assuming that only eight bits are needed to store the indices). Wires available on the bus are as follows:

ABus the address lines

DBus the data lines

R/$\overline{\text{W}}$ memory read/write control

BReq request control of the bus

BGrant granting control of the bus

IO/$\overline{\text{M}}$ IO/memory access indication

With state machine diagrams, describe communication protocols as follows:

(a) between the monitoring program and the input device for one time initialization of the latter by the former and subsequent synchronization on buffer full / empty conditions, and **(5)**

(b) between the CPU and the input device which will enable the latter to write bytes of data into the circular buffer. **(5)**

C.4 (a) Suppose you are given a CPU with 16 GPRs `R0 -- R15` and only one ALU for computing all addresses (of instructions and data) and carrying out operations on data. Let the memory word be 4B. For the instruction

`beq Rsrc1, Rsrc2, offset`, with RTL interpretation

`if Rsrc1 = Rsrc2 then PC <= PC + 8 + offset else PC <= PC + 8`,
where `Rsrc1, Rsrc2` are GPRs and the instruction takes two words
with the second word holding the `offset`,

(i) Draw the part of the data path of the CPU that supports fetch (of all instructions) and the execution phase of the above instruction. **(3)**

(ii) Give the RTL micro-operation sequence of the fetch and execution phases of the instruction. **(3)**

(b) Consider a 4-way set-associative 64KB data cache with 128B block size and a 32-bit address main memory.

(i) Find the total cache size and the number of comparators needed for accessing the cache. Show the calculations to justify your answer. **(2)**

(ii) Give a neat schematic diagram depicting the cache access mechanism. **(2)**

C.5 (a) Draw the process state transition diagram of a non-preemptive OS. Mark all states and edges properly. **(3)**

(b) In an online noticeboard system, sellers post items they wish to sell. Buyers look up the noticeboard to see the items posted, and then contact the seller offline if they are interested in any item. The system is used very heavily with a large number of buyers and sellers connecting to the system most of the time. Every request over the network to the system (buyer or seller) generates one separate process on the system that does the work (you do not have to worry about network congestion or the problem of creating too many processes, etc.). To keep the design simple, the online system designer needs to enforce that no seller updates the noticeboard while any buyer is reading the noticeboard, and no two sellers update the noticeboard at the same time. Can you write the pseudocode for the buyer and seller processes to do this? First identify the requirements that your solution should satisfy (or guarantees that your solution should give to the buyers or sellers) clearly, then write the pseudocode to implement them using semaphores. **(7)**

C.6 (a) Consider a demand paged system with a 64-bit address space, 8 KB page size, and 128 GB RAM. What would be the total size of the page table(s) for one process if we use (i) a 1-level page table, (ii) a 2-level page table? Show the calculations. **(3 + 4)**

(b) How does the OS ensure that a page replacement algorithm does not choose a page containing critical OS code for replacement (for example, think what can happen if the page replacement algorithm chooses to swap out the page that contains the page replacement code!)? **(3)**