PhD Qualifier Examination

Department of Computer Science and Engineering

Date: 28-Mar-2019 **Maximum Marks:** 100

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

exp − 127

(0.1)

exp 127

(0.1N) B

Group A

A.1 Write only the answers to the following parts. No justifications are needed.

(a) For the functions $f(n) = \dfrac{5n}{7n+6}$ and $g(n) = 1 + \dfrac{2\log n}{3n}$, 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) Let the running time $T(n)$ of an algorithm satisfy the recurrence $T(n) = T(\sqrt{n}) + 1$. Which of the following is true about $T(n)$? **(3)**

(i) $T(n) = \Theta(\sqrt{n})$ (ii) $T(n) = \Theta(\log n)$ (iii) $T(n) = \Theta(\log\log n)$ (iv) $T(n) = \Theta(1)$

(c) Let A be an array storing a (binary) max-heap in the contiguous representation. We store the reverse of A in an array B. Give an example of a max-heap stored in A such that the array B storing the reverse of A is not be a min-heap in the contiguous representation. **(3)**

(d) What is the worst-case running time (in the big-O notation) of searching for an element in a hash table with open addressing and storing n elements? **(2)**

A.2 You are given a binary search tree T of height h. Propose an $O(h)$-time algorithm to solve the following problem: Given x, find the key stored in T, that is closest to x. If x is already stored in T, then the output is x itself. If x is not stored in T, you need to find y for which $|x - y|$ is the smallest among all keys y stored in T. A tie (if it arises) may be broken arbitrarily. **(10)**

A.3 A secret device contains an array A of n distinct integers. Your task is to create an array B in the local memory of your machine to store a sorted version of A. You cannot see the integers in the device. All you can do is to make queries to the device of the form $query(L, R, i)$ with $L \leqslant R$. The device returns the i-th smallest integer among those elements of A, that lie between L and R (both inclusive). If A contains less than i elements in the range $[L, R]$, then $query(L, R, i)$ returns an error. If A contains t numbers in the range $[L, R]$, the device takes $O(t)$ time to answer this query (or notify error).

For example, $query(-\infty, +\infty, n)$ returns the maximum element of A (assume that you know the size n of A). If $A = (29, 10, 9, 17, 7, 24, 32, 12, 16, 43)$, then $query(10, 25, 3)$ returns 16 (the third smallest element among $10, 17, 24, 12, 16$), whereas $query(10, 25, 6)$ encounters an error (A contains only five elements in the queried range).

Propose an $O(n \log n)$-time algorithm to create the sorted array B by making queries of the given type. **(10)**

A.4 Let S be a string of length n. Your task is to construct a subsequence T of S (note that a subsequence of a string is not required to be contiguous). For every symbol you include in T, you get a profit p_1. For every repetition of symbols in consecutive positions in T, you have to pay a penalty p_2. For example, if $S = cabcacbbabcbcbaa$, and you choose $T = accbabbba$ (the underlined symbols of S), then your total profit is $9p_1$, and your total penalty is $3p_2$. Assume that $p_2 > p_1 > 0$. Propose an $O(n)$-time algorithm to maximize your gain (that is, your total profit minus your total penalty). **(10)**

A.5 Let $G = (V, E)$ be a tree (that is, an undirected connected acyclic graph) given in the adjacency-list representation. You are given a vertex $r \in V$. G can be viewed as a tree rooted at r. The level of a vertex $v \in V$ is the distance of v from r in G. Propose an efficient algorithm to compute the levels of all the vertices of G. What is the running time of your algorithm? **(7 + 3)**

A.6 Consider a Butterfly pattern for a given positive integer n, where the two wings of the butterfly will be generated through the star ($*$) symbols. For a given n, the left wing starts with $2n$ stars in the first column, followed by $2n-2$ stars in the second column, and so on, and goes up to 2 stars. The right wing starts with 2 stars, followed by 4 stars, and so on, and goes up to $2n$ stars.

Example Butterfly patterns for $n = 5$ and $n = 6$ are given below.

```
*                 *
*  *           *  *
*  *  *     *  *  *
*  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *
*  *  *  *     *  *  *  *
*  *  *           *  *  *
*  *                 *  *
*                       *
```

$n = 5$

```
*                                *
*  *                          *  *
*  *  *                    *  *  *
*  *  *  *              *  *  *  *
*  *  *  *  *        *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *        *  *  *  *  *
*  *  *  *              *  *  *  *
*  *  *                    *  *  *
*  *                          *  *
*                                *
```

$n = 6$

Write a C function `void butterfly (int n)` to print the Butterfly pattern for the given input n. **(10)**

A.7 Consider an array $A[\,]$ of N integers, where $N = 2k+1$ is odd. Write a C program to V-sort this array, that is, to ensure that

$$A[k] \leqslant A[k-1] \leqslant A[k+1] \leqslant A[k-2] \leqslant A[k+2] \leqslant A[k-3] \leqslant A[k+3] \leqslant \cdots .$$

Note that $A[k]$ is the middle element of the array (assuming zero-based indexing). For example, if $A[\,] = \{2,6,12,5,18,22,9,20,32\}$, then the V-sorted array will be $\{22,18,9,5,2,6,12,20,32\}$. **(10)**

A.8 Let $z = x+iy$ be a complex number. The absolute value of z is given by $|z| = \sqrt{x^2+y^2}$. Consider a doubly-connected linked list of complex numbers, where the nodes are sorted in the ascending order by the absolute values of the complex numbers stored at the nodes.

Consider that every complex number is represented as a C structure `complex` which contains a floating-point variable `real` for storing the real part and another floating-point variable `imaginary` to store the imaginary part. Every node of the linked list contains a variable of the structure `complex`, and two links, one pointing to the previous node, and another pointing to the next node.

(a) Write a C structure to store a node of this linked list. **(3)**

(b) Write a C function to insert a new complex number to an existing sorted linked list, based on the sorting criteria explained above. **(7)**

Group B

B.1 (a) A binary string (like the string 101101 of length six) which is unchanged upon reversing order is called a palindrome. Suppose that a binary string of length six is randomly generated, with 0 and 1 equally likely for each of the six positions in the string. What is the probability that the string is a palindrome if the first and the sixth bits **(5)**

(i) are both 1,

(ii) are the same?

(b) A carton contains 9 computer chips, four of which are faulty, and the rest of which are good. If three chips are taken from the box one at a time (without replacement), find the probability that the three chips are alternatively either 'good, faulty, good' or 'faulty, good, faulty'. **(5)**

B.2 (a) Consider the two sets $X = \{1,2,3\}$ and $Y = \{a,b,c\}$. State whether the following are true or false. Justify your answer. Here, $X \times Y$ denotes the Cartesian product of X and Y, and $|X|$ denotes the cardinality of the set X. **(6)**

(i) $X \times Y = Y \times X$

(ii) $X \times X = Y \times Y$

(iii) $|X \times Y| = |Y| \cdot |X|$.

(b) If $|X| = n$, and if $P(X)$ denotes the power set of X, prove that $|P(X)| = 2^n$. **(4)**

B.3 The following recursive algorithm evaluates the polynomial $p(x) = \sum_{k=0}^{n} c_k x^{n-k}$ at a point t.

```
Input: The sequence of coefficients c0, c1, c2, ..., cn, the value t, and n
Output: p(t)
poly(c,n,t)
{
    if (n = 0) return c0
    return t × poly(c,n − 1,t) + cn
}
```

Let b_n be the number of multiplications required to compute $p(t)$ by this algorithm.

(a) Find a recurrence relation and an initial condition for the sequence $\{b_n\}$ with $n \geqslant 0$. **(4)**

(b) Compute b_1, b_2. **(2)**

(c) Solve the recurrence relation of Part (a). **(4)**

B.4 Prove or disprove the following statements.

(a) Let L be a regular language, and $L_1 \subseteq L$. Then L_1 must also be regular. **(3)**

(b) Let L_2 be a language that can be written as the union of two languages L_3 and L_4, that is, $L_2 = L_3 \cup L_4$. For L_2 to be regular, both L_3 and L_4 have to be regular. **(3)**

(c) Let L_5 be a language that is not regular. Then the complement $\overline{L_5}$ is also not regular. **(4)**

B.5 (a) Construct a DFA for the set of all strings that start with ab and end with ba. The alphabet is $\{a,b\}$. **(4)**

(b) Construct a PDA for the language L_6, where $\#_s(w)$ denotes the number of the symbol $s \in \{a,b\}$ in the string w. State the logic clearly, and provide a graphical description of the PDA. **(6)**

$$L_6 = \left\{ w \in \{a,b\}^* \mid 2\#_a(w) \neq 3\#_b(w) \right\}$$

B.6 (a) Provide a context-free grammar for the language $L_7 = \left\{ a^i b^j c^k \mid i + j > k \right\}$. **(6)**

(b) Give a regular expression for the language $L_8 = \left\{ a^n b^m \mid n + m \text{ is odd} \right\}$. **(4)**

Group C

C.1 (a) Prove the following identity algebraically: $A + A'B = A + B$, without using the truth table method. **(2)**

(b) Minimize the 4-variable function

$$F(A,B,C,D) = \Sigma(2,3,5,10) + \Sigma_\phi(1,6,11,14)$$

using the Karnaugh-map method, and hence realize the function using NAND gates only. Here, Σ denotes the true minterms, and Σ_ϕ denotes the don't-care minterms. **(4)**

(c) Write down the sum-of-products expression for the sum output of a full adder. Hence implement the function using a 4-to-1 multiplexer and a NOT gate. **(4)**

C.2 (a) Compute 25 – 36 using 6-bit 2's complement representation. Show all steps of the computation. **(3)**

(b) Show a typical 32-bit floating-point representation of numbers. Comment on the range and precision of the numbers that can be represented. **(4)**

(c) Draw the schematic diagram of a modulo-20 binary counter using J-K flip-flops. **(3)**

C.3 (a) Draw the schematic diagram of a 4-bit adder/subtractor using full adders and gates as basic building blocks. Make relevant assumptions. **(3)**

(b) With the help of diagrams, explain the following addressing modes: **(3)**

(i) indexed addressing,
(ii) relative addressing,
(iii) base addressing.

(c) In a single-bus processor architecture, with general-purpose registers, program counter (PC), memory address register (MAR), memory data register (MDR), and instruction register (IR), show the register-transfer level operations required to execute the following instruction: **(4)**

ADD R2,M(R5) // R2 = R2 + Mem[R5]

C.4 (a) Multiple I/O devices are required to be connected to a single interrupt line of the processor. Suggest one method by which interrupt priorities can be resolved. **(4)**

(b) Consider a 1Mbyte main memory with 20-bit memory address. Consider a set-associative cache memory organization with the following parameters: cache block size of 32 bytes, 64 sets, and 4 blocks per set. Show a schematic diagram of the memory organization, and hence show how the bits of the memory address will be divided into TAG, SET and WORD fields. **(3)**

(c) In a two-level cache system, the hit time of the first cache L_1 is 20 nsec and that of L_2 is 40 nsec. If the hit rate of L_1 is 80% and that of L_2 is 90%, and the miss penalty on an L_2 cache miss is 200 nsec, calculate the average memory access time. Assume that the hit rate of L_2 is computed based upon the number of requests that hit L_2 (that is, upon an L_1 miss). **(3)**

C.5 (a) Define a semaphore. Explain how you can make the semaphore operations atomic in a multiprocessor system (clearly state any assumptions you make). **(2 + 3)**

(b) Suppose that a process P is running on a uniprocessor system with an OS having round-robin scheduling. Explain clearly the sequence of events (from an OS point of view) when P does a *scanf*() to read something from the keyboard. **(5)**

C.6 (a) What happens to the TLB on a context switch? **(2)**

(b) Suppose that a process P hits a page fault and there are no free page frames. The page-replacement algorithm is now invoked. The algorithm chooses a page containing data of another process Q for replacement. Clearly list the steps that will be taken by the OS for the page fault to be handled. You should refer to relevant OS data structures, if any, for each step in your answer. **(5)**

(c) Suppose that there are two processes P and Q in a system with round-robin scheduling. Both P and Q have already run for two time quanta each, and currently P is running in its 4th time quanta. When the time quanta of P expires and Q is scheduled, explain how the OS will start using Q's page table instead of P's. **(3)**

— END —