PhD Qualifier Examination
Department of Computer Science and Engineering

Date: 31-Oct-2017 **Maximum Marks:** 100

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

Group A

A.1 In the following parts, write only the correct answers. There is no need for explanation.

(a) Consider the two functions of integers $n \geqslant 0$: $f(n) = n\log^2 n$, and $g(n) = 5n\log\log n$. Which of the following statements is correct? **(2)**

(i) $f(n) = O(g(n))$
(ii) $g(n) = O(f(n))$
(iii) $f(n) = \Theta(g(n))$
(iv) None of the above

(b) Suppose that most of the elements in an array A of n integers are 0. There are only a constant number of non-zero elements in A. What are the worst-case running times in the big-theta notation of the following algorithms to sort A? **(3)**

(i) Insertion sort
(ii) Merge sort

(c) Let H be a max-heap of distinct elements. Mention which of the following statements is/are true and which is/are false. **(3)**

(i) The second maximum is always stored in a child of the root of H.
(ii) The minimum value is always stored in a leaf of H.
(iii) The second minimum value is always stored in a leaf of H.

(d) What is the running time of Dijkstra's single-source-shortest-path algorithm on a tree consisting of n vertices? Assume that the links of the tree are presented to you in the form of an adjacency matrix, and that a heap is used for choosing vertices during the working of the algorithm. **(2)**

A.2 Let A be an unsorted array of $n = 2k$ integers. It is given that exactly half (that is, k) of the elements of A are positive, and the remaining k elements are negative. Your task is to change A *in place* such that the positive and negative elements of A are positioned alternately, and the modified array starts with a positive element. For example, if $A = (2, 6, 4, -7, -3, 5, -9, -1)$, it may be changed to $(2, -3, 4, -7, 6, -9, 5, -1)$. You do not have to preserve the order of the positive or negative elements as they appear in the original array. Propose an $O(n)$-time algorithm to solve the problem. Your algorithm must use only a constant amount of extra space. In particular, no extra arrays may be used. **(10)**

A.3 Consider the following divide-and-conquer algorithm for sorting an array A of n integers. If $n = 0, 1$, do nothing. If $n = 2$, swap the elements if necessary. For $n \geqslant 3$, let $t = \lceil 2n/3 \rceil$. Recursively sort the first t elements of A. Then, recursively sort the last t elements of A. Finally, sort the first t elements of A recursively. All the three recursive calls sort the chunks of A in place.

(a) Prove that this algorithm correctly sorts A. **(7)**

(b) Derive the running time of this algorithm. **(3)**

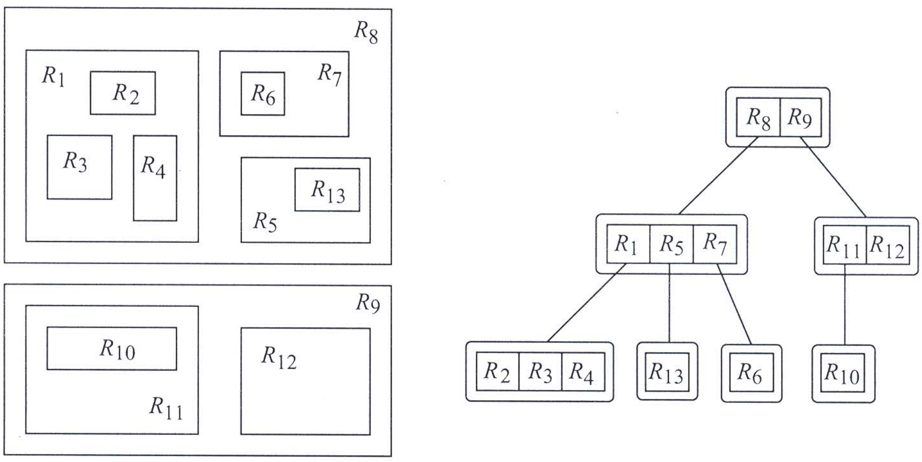
A.4 Let s and t be two vertices in a connected undirected graph $G = (V, E)$. A particle sits at vertex s, and wants to reach vertex t. As the particle hops more and more along the edges, it gets more and more tired. In the first hop, it expends one unit of energy. In the second hop, it expends two units of energy. In the third hop, it expends four units of energy. In general, in the i-th hop, the particle expends 2^i units of energy. Propose an efficient algorithm for the particle to reach t from s after expending the minimum possible amount of energy. What is the running time of your algorithm? **(7+3)**

A.5 You are given n distinct rectangular regions $R_1, R_2, \ldots, R_n$ in the plane. For every two regions R_i, R_j, exactly one of the following conditions holds.

(1) R_i is completely contained in R_j.
(2) R_j is completely contained in R_i.
(3) R_i and R_j are disjoint (that is, non-overlapping).

You are given a subroutine that, upon the input of two regions, outputs which of the above three conditions holds. You do not have to propose an algorithm to implement this subroutine.

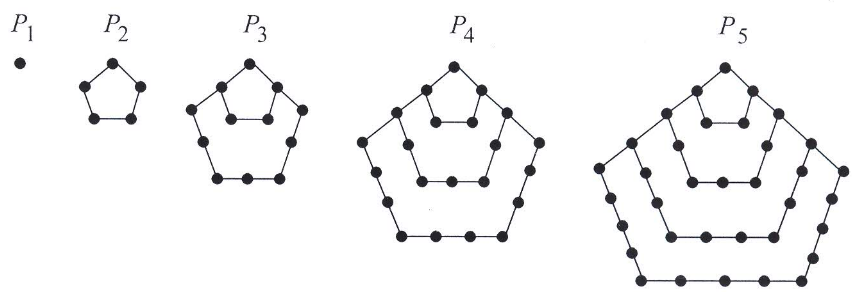
The regions are to be stored in a tree called an R-tree. Each node of the tree contains one or more regions. The root consists of those regions which are not contained in any other regions. There are links from regions to nodes. If R_i is the smallest region to contain R_j, then there is a link from R_i to the node containing R_j. This construction is demonstrated in the figure below.



(a) The regions (b) The R−tree

Propose an algorithm to insert a new region R in an R-tree T. Assume that R is not already stored in T, and satisfies the three relations stated above with the regions already present in T. You do not have to specify the data structures used for storing T. Just mention how the insertion procedure changes the structure of the tree. **(10)**

A.6 Consider the recursive way of building a stack P_n of pentagons as shown in the following figure.

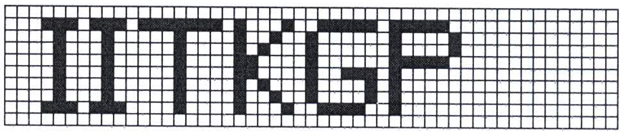


Notice that the outermost pentagon in P_n contains n dots on each side, the second outermost pentagon in P_n contains $n-1$ dots on each side, and so on. The number p_n of dots in P_n is called the n-th pentagonal number. For example, $p_1 = 1$, $p_2 = 5$, $p_3 = 12$, $p_4 = 22$, $p_5 = 35$, and so on.

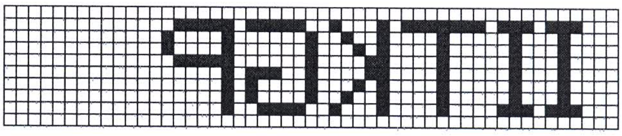
(a) Write a recursive C function that, upon the input of n as the parameter, returns the n-th pentagonal number p_n. **(8)**

(b) What is the time complexity of your function? **(2)**

A.7 An image is specified by a two-dimensional array of pixels. For simplicity, assume that each pixel can have only two levels: 0 means white, and 1 means black. Your task is to flip an image horizontally. This is demonstrated in the following figure. Each small square represents a pixel in the image.



(a) An image storing a text



(b) The horizontal flip of the image

Write a C function to flip an image horizontally. The flipping is to be done *in place*, that is, in the input two-dimensional array itself. Do not use any extra array. **(10)**

A.8 In a usual binary search tree T, the insertion of a key already present in T produces no effect. Suppose instead that we maintain in each node of T a linked list storing all insertion instances of the same key, where each insertion instance is specified by a timestamp. During the insertion of a key k with timestamp t, one first checks whether k is present in the tree. If present, t is inserted at the beginning of the linked list holding the earlier insertion instances of k. Otherwise, a new node is created to store k and a single-element linked list storing t.

(a) Write a user-defined C data type to store a node in this binary search tree. **(3)**

(b) Write a C function to insert in this binary search tree a key with a timestamp using the modified insertion procedure stated above. **(7)**

Group B

B.1 (a) Consider the functions $f, g : \mathbb{R} \to \mathbb{Z}$ defined as follows (where $\mathbb{R}$ is the set of real numbers, and $\mathbb{Z}$ is the set of integers):

$$f(x) = \begin{cases} x & \text{if } x \in \mathbb{Z}, \\ \lfloor x \rfloor + 1 & \text{if } x \in \mathbb{R} \setminus \mathbb{Z}, \end{cases}$$

and $g(x) = \lceil x \rceil$ for all $x \in \mathbb{R}$. Prove that $f(x) = g(x)$. **(4)**

(b) Closely related to the Fibonacci numbers is the sequence known as *Lucas numbers*, which is defined recursively as:

$$\begin{aligned} L_0 &= 2 \\ L_1 &= 1 \\ L_n &= L_{n-1} + L_{n-2} \quad \text{for } n \geqslant 2,\ n \in \mathbb{Z}^+ \end{aligned}$$

Prove that $L_1^2 + L_2^2 + L_3^2 + \cdots + L_n^2 = L_n L_{n+1} - 2$. **(6)**

B.2 (a) Let S_n denote the number of n-bit strings over $\{0,1\}$ that do not contain the pattern `111`. Find a recurrence relation for the sequence $S_0, S_1, S_2, \ldots, S_n, \ldots$, and the initial conditions that define the sequence. **(4)**

(b) Solve the following recurrence relation: **(6)**

$$\begin{aligned} a_0 &= 1 \\ a_1 &= 2 \\ a_n &= -2na_{n-1} + 3n(n-1)a_{n-2} \quad \text{for } n \geqslant 2. \end{aligned}$$

B.3 (a) Let X be a non-empty set. Define a relation R on $\mathscr{P}(X)$, the power set of X, as:

$$(A, B) \in R \text{ if and only if } A \subseteq B$$

Is this relation reflexive? Symmetric? Antisymmetric? Transitive? **(4)**

(b) A bag contains 5 white and 7 black balls. If two balls are drawn at random, what is the probability that one is white and the other is black in the following two cases? **(6)**

(i) The two balls are drawn together.
(ii) The balls are drawn one by one with replacement.

B.4 Prove or disprove the following.

(a) $L_1 = \{a^n \mid n \text{ is a perfect square } \geqslant 0\}$ is a regular language. **(4)**

(b) Let L_2 be the language of all palindromes over $\Sigma = \{a, b\}$, which is known to be non-regular. Then, *any* infinite subset L_3 of L_2 such that every string in L_3 contains at least one a and at least one b is also non-regular. **(4)**

(c) There is a language L_4 over the alphabet $\Sigma = \{a, b\}$ for which $(\overline{L_4})^* = \overline{L_4^*}$. **(2)**

B.5 (a) Build a CFG to accept the following language: **(5)**

$$L_5 = \{a^i b^j c^k \mid i > 2j + 3k\}$$

(b) Construct a DFA that accepts the set of all odd-length strings that contain the substring ab. **(5)**

B.6 (a) Consider the grammar:

$$\begin{aligned} E &\rightarrow E + T \mid T \\ T &\rightarrow T \times F \mid F \\ F &\rightarrow (E) \mid a \end{aligned}$$

Show the derivation and the parse tree of the string `((a))` using this grammar. **(4)**

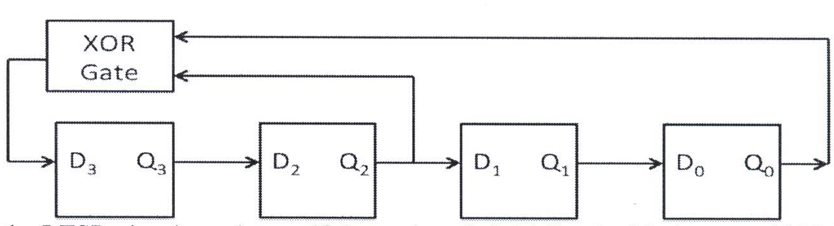
(b) Construct a Push-Down Automata for the following language over $\Sigma = \{0, 1\}$:

$$L_6 = \{w \in \{0,1\}^* \mid \#0(w) \geq \#1(w)\}$$

where $\#0(w)$ and $\#1(w)$ denote the number of 0's and the number of 1's in the string w, respectively. **(6)**

Group C

C.1 A linear feedback shift register (LFSR) is a modified shift register circuit consisting of D-type flip-flops as shown in the figure below, where an exclusive-OR gate with selected tap points as inputs is used to feed the serial-in data into the shift register. The exclusive-OR gate can have any number of inputs.



(a) For the LFSR circuit as shown, if the register is initialized with the state 1000 (in binary, from left to right), determine the number of distinct states the register will go through before repeating (that is, number of states in a cycle). **(4)**

(b) For an n-bit LFSR, can all the 2^n distinct states lie in a cycle? Justify your answer. **(3)**

(c) An n-bit LFSR is said to be a maximal-length LFSR if the number of distinct states in a cycle is $2^n - 1$. If the number of inputs to the exclusive-OR gate is odd, justify that the LFSR can never be maximal-length. **(3)**

C.2 (a) Which of the following constitute functionally complete set(s), assuming that constant 0 and constant 1 inputs are available? **(3)**

(i) {NOT, XOR} (ii) {AND, XOR} (iii) {2-to-1 MUX}

(b) Implement the following function using an 8-to-1 multiplexer and extra gates if required: **(3)**

$$F(A,B,C,D) = A'BD + AB'C' + BC$$

(c) Consider a sequential circuit that has a serial input X and a serial output Z. The input bits are applied in synchronism with a clock signal. The output Z will become 1 whenever the bit pattern `101010` appears in the input stream (may be overlapping patterns). For example, if the input pattern is `001100101010100`, the corresponding output pattern will be `000000000001010`. Draw the state transition diagram of the FSM, and realize it using basic gates. **(4)**

C.3 (a) What do you mean by sign extension of a 2's complement number? Illustrate the concept by representing -35 in 6-bit 2's complement, and then extend it to 32-bit representation. **(2)**

(b) It is required to design a 16-bit adder/subtractor, with two 16-bit inputs A and B, and a 16-bit result Z. A control signal *mode* determines the operation to be performed; if $mode = 1$ we perform addition; if $mode = 0$, we perform subtraction. Show the register-transfer level datapath for the implementation. You are allowed to use only a single adder or a subtractor in the design. **(5)**

(c) Consider the instruction set architecture of a general-purpose machine. Suppose that a total of 60 control signals are present, out of which 15 are mutually exclusive (that is, no more than one of these control signals are active at a time) while the rest are not. What will be the minimum number of bits required in the control word for microprogrammed implementation? **(3)**

C.4 (a) Consider a 2-level memory hierarchy with separate instruction cache (I-cache) and data cache (D-cache) in Level 1, and main memory in Level 2. The clock-cycle time is 1 nanosecond. The miss penalty is 20 clock cycles for both read and write. 2% of the instructions are not found in I-cache, and 10% of data references are not found in D-cache. 25% of the total memory accesses are for data, and cache access time (including hit detection) is 1 clock cycle. What will be the average access time of the memory hierarchy? **(3)**

(b) Consider a computer system with the following characteristics: main memory size of 1 Mbyte, word size of 1 byte, block size of 16 bytes, and cache size of 64 Kbytes. For the hexadecimal memory addresses `F2210` and `DB3CE`, give the corresponding tag, cache set, and offset values for a two-way set associative cache. **(2)**

(c) A processor has a single interrupt input, and it is required to connect 4 peripheral devices that will be transferring data to the processor in interrupt-driven mode. Clearly suggest a scheme for doing this (with a detailed schematic diagram), explaining how the appropriate interrupt handler gets invoked upon receipt of an interrupt. **(5)**

C.5 (a) Consider an OS using a round-robin scheduler with a fixed time quantum per process. Suppose that two processes A and B are run on the OS in a system with a single CPU, with each of the processes running the following C program:

```
#include <stdio.h>
int main()
{
    while(1);
}
```

Suppose that process A is currently running on the CPU. Explain clearly step by step what happens when A's time quantum expires. Your answer should attempt to give sufficient details of each step to indicate how each step is implemented. **(5)**

(b) Is it possible for a process to go from running to blocked state even when there is no input or output required? If yes, give an example. If no, justify briefly why not. **(2)**

(c) Outline how can you make the `wait()` and `signal()` operations on a semaphore atomic in a system with a single CPU. **(3)**

C.6 (a) Consider an OS using inverted page tables for virtual memory management, with 64-bit virtual address, 16 KB page size, and 128 GB RAM. The OS can support a maximum of 8192 processes at any one time. Consider that 10 processes are currently running in a system with the OS. What is the total size of the inverted page table if memory is aligned at byte boundaries, and the additional memory space for bits like valid/dirty etc. are negligible and can be ignored? Show all calculations clearly. **(6)**

(b) In a demand-paged OS with a 1-level page table, how can it be prevented that the pages of the page table are not themselves paged out? **(2)**

(c) When does a CPU go to its privileged state? **(2)**
