Pritam

# PhD Qualifier Examination

## Department of Computer Science and Engineering

**Date:** 30-Mar-2016 **Maximum Marks:** 100

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

---

### Group A

**A.1** Write only the answers of the following questions. No need to write any explanation. **(2 × 5)**

(a) An undirected graph $G$ having $n$ vertices and $m$ edges is represented by an adjacency matrix $M$. What is the space complexity of $M$?

(b) For sorting $n$ 4-bit binary numbers, where $n$ is very large (say in millions), which sorting algorithm would be most efficient? What is its time complexity?

? (c) Match the following functions in the left column with their corresponding asymptotic upper bounds in the right column.

| | |
|---|---|
| A) $f(n) = \sum_{i=1}^{n} \frac{n}{i}$ R | P) $O(\log n)$ |
| B) $f(n) = \sum_{i=1}^{n} \frac{1}{i^2}$ S | Q) $O(n^2)$ |
| C) $f(n) = f(n/2) + f(n/4) + 13$ P | R) $O(n \log n)$ |
| D) $f(n) = 4f(n/2) + n/2$ R | S) $O(1)$ |

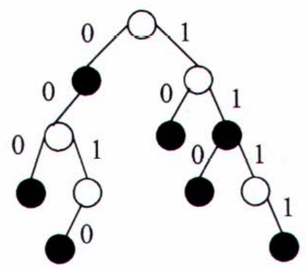
? (d) $A$ is a 1D (unsorted) array containing $n$ distinct numbers. Write the worst-case time complexity of the best possible algorithm for finding an element $x$ in $A$ such that at least $n/10$ elements of $A$ are less than $x$ and at least $n/10$ elements of $A$ are greater than $x$. Assume that $n$ is sufficiently large. $O(n)$

(e) $A$ and $B$ are two $n$-bit binary strings, indexed from 1 to $n$. A function $\phi$ is defined on them as follows.

$$\phi(A,B,i) = \begin{cases} 0 & \text{if } i = 0 \\ \phi(A,B,i-1)+1 & \text{if } A[i] \neq B[i] \text{ and } 1 \leq i \leq n \\ \phi(A,B,i-1) & \text{otherwise.} \end{cases}$$

Write the values of $\phi(A,B,3)$ and $\phi(A,B,4)$ if $A = 011$ and $B = 101$.

**A.2** You have a set of pairwise distinct binary strings. Let $n$ be the sum of the lengths of the strings. You can store the strings in a rooted binary tree with at most $n+1$ nodes. Some nodes in the tree (including all the leaf nodes) are marked. Imagine that each left-child link is labeled by 0, and each right-child link by 1. The tree is said to store a binary string $s$ if there exists a marked node $v$ in the tree such that the unique path from the root to $v$ is labeled by the symbols of $s$. The following figure exemplifies such a tree with the marked nodes shown as solid circles. The binary strings stored in this tree are: 0, 000, 0010, 10, 11, 110, and 1111. The tree dose not store the strings 001 (the path ends in a non-marked node) and 010 (there is no path from the root with this label).



(a) Propose an efficient algorithm to insert a binary string $s$ in the tree. If $s$ already resides in the tree, no change should be made to the tree. **(5)**

(b) Propose an $O(n)$-time algorithm to print all the strings stored in the tree in lexicographic ordering. **(5)**

**Note:** Write your algorithms as pseudocodes or C functions. Do not only supply English descriptions.

$n^2c_n \mid n^2c_1 + \cdots n^2c_{n-1}$ .

A.3 (a) We store $n$ keys in a hash table of size $n^2$ using a hash function. Assuming that the hash function generates uniformly random output in the range $[0, n^2 - 1]$, prove that the probability is less than $1/2$ that there are any collisions. **(5)**

(b) A $d$-ary heap is like a binary heap with the exception that non-leaf nodes have (at most) $d$ children instead of two. How will you store such a heap in an array? Analyze the running time of an efficient insert operation in a $d$-ary max-heap in terms of $d$ and $n$. **(5)**

A.4 The longest increasing subsequence (LIS) problem is to find a longest subsequence (not necessarily contiguous) of a given sequence (of integers) so that the elements in the subsequence are in sorted (ascending) order. For example, for the sequence 0, 8, 4, 12, 2, 10, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15, a longest increasing subsequence is 0, 2, 6, 9, 11, 15.

(a) Propose an efficient algorithm to solve this problem. (**Hint:** Use an array $L$ such that $L[i]$ stores the LIS *ending* at position $i$.) **(8)**

(b) Deduce the running time of your algorithm. **(2)**

? A.5 A community consists of $n$ members. It is said to be *well-built* if each of its members votes directly or indirectly for each other. The *direct votes* are available in a 2D binary array $M$. $M[i][j] = 1$ if the $i$th member *votes directly* for the $j$th member, which however does not imply $M[j][i] = 1$; that is, $M[i][j] = M[j][i] = 1$ if and only if the $i$th and the $j$th members vote directly for each other.

The $i$-th member *votes indirectly* for the $j$-th member if there exists a sequence $\langle i, k_1, k_2, \ldots, k_t, j \rangle$ of members such that $t \geq 1$ and $M[i][k_1] = M[k_1][k_2] = \cdots = M[k_t][j] = 1$.

Given $M$, we want to check whether the community is well-built. Suggest an efficient algorithm for this, and explain its time complexity. Strongly connected $O(n^3)$ **(7+3)**

A.6 The Fibonacci numbers are recursively defined as: $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$. Any positive integer $N$ can be represented as

$$N = F_{n_k} + F_{n_{k-1}} + \cdots + F_{n_2} + F_{n_1} + F_{n_0},$$

where $n_0 \geq 2$ and $n_i \geq n_{i-1} + 2$ for $i = 1, 2, \ldots, k$. It can be proved that such a representation of $N$ is unique. It is called the *Zeckendorf representation* of $N$. Write an *efficient* C function that takes a positive integer $N$ as input, and computes and prints the Zeckendorf representation of $N$. **(10)**

A.7 You are given a string of opening and closing parentheses, such as `((())((())))`.

(a) Write a function to return the maximum depth of parentheses (four in above example), provided that the parentheses are balanced. If the parentheses are unbalanced, your function should return $-1$. **(8)**

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

A.8 We consider a Disjoint Interval Tree $T(l, u, h)$ as a binary tree of height $h$ with the following conditions.

- Any node $a$ is labeled with an integer interval $[l_a, u_a]$ with $l_a \leq u_a$.
- For any node $a$ in the tree, any node $b$ occurring in the left subtree of $a$ satisfies $u_b \leq l_a$.
- For any node $a$ in the tree, any node $b$ occurring in the right subtree of $a$ satisfies $u_a \leq l_b$.
- If $t$ is the leftmost node in $T$, then $l_t = l$ and similarly, if $t$ is the rightmost node in $T$, then $u_t = u$.
- Every real number $x$ in the real interval $[l, u]$ belongs to exactly one interval stored in some node of the tree; or, $x$ belongs to exactly two intervals stored in two nodes of the tree if $x$ is the upper limit of one interval and also the lower limit of the other interval.

(a) Propose a data type to store a node of a Disjoint Interval Tree. **(3)**

(b) Write a C function *create_interval_tree(l,u,h)* which creates a *complete* Disjoint Interval Tree with nodes storing intervals of *almost* equal lengths. **(7)**

## Group B

✓B.1. (a) Show that if $r$ is an irrational number, there is a unique integer $n$ such that the distance between $r$ and $n$ is less than $\frac{1}{2}$. **(5)**

. (b) Let the *sequence* $T$ be defined by $T_1 = T_2 = T_3 = 1$ and $T_n = T_{n-1} + T_{n-2} + T_{n-3}$ for $n \geq 4$. Prove by induction that $T_n < 2^n$ for all positive integers $n$. **(5)**

✓B.2. (a) Let $f$ be a function from the set $A$ to the set $B$. Let $S$ and $T$ be subsets of $A$. Show that:

i) $f(S \cup T) = f(S) \cup f(T)$ **(2)**

ii) $f(S \cap T) \subseteq f(S) \cap f(T)$ **(2)**

. (b) Let $A = \{1,2,3,4\} \times \{1,2,3,4,6\}$. A relation on $A$ is defined by $(x,y)\ R\ (s,t)$ if and only if $xt = ys$.

i) Show that $R$ is an equivalence relation. **(3)**

ii) For each equivalence class, list one member belonging to it. 12 **(3)**

✓B.3. (a) A computing center has three processors that receive $n$ jobs, with the jobs assigned to the processors purely at random so that all of the $3^n$ possible assignments are equally likely. Find the probability that exactly one processor has no job. $\frac{2^n-2}{3^{n-1}}$ **(5)**

. (b) Suppose you get two times as many spam e-mail messages as non-spam e-mail messages. Suppose that the probability that a spam e-mail contains the word *free* is $1/3$, and the probability that a non-spam e-mail contains the word *free* is $1/20$. If you receive an e-mail containing the word *free*, what is the probability that the e-mail is spam? $40/43$ **(5)**

B.4 (a) Let $L_1$ and $L_2$ be two infinite languages, defined over the alphabet $\{a,b\}$, satisfying $L_1 \cap L_2 = \phi$ and $L_1L_2 = L_2L_1$. If such a language pair exists, give an example. If not, you must prove it. **(6)**

(b) Consider the following grammar:

$$
\begin{aligned}
S &\rightarrow abScB \mid \varepsilon \\
B &\rightarrow bB \mid b
\end{aligned}
$$

What language does it generate? **(4)**

B.5 (a) Construct an NFA to accept the regular expression $b(((ba)^* + bbb)^* + a)^*b$, such that the number of states are as minimum as possible. You should not use $\varepsilon$-transitions. (**Hint:** The required number of states is $\leq 5$. You will be penalized if the number of states in your NFA increases beyond the required number.) **(5)**

(b) Provide a CFL for the following language: $L_3 = \{a^ib^jc^kd^l \mid i + j = k + l\}$. **(5)**

✓B.6 Prove or disprove the following statements:

. (a) Any infinite subset of the language $L_4 = \{a^nb^m \mid n = m \text{ or } n = 2m\}$ has to be non-regular. **(6)**

. (b) Given a language $L$, define:

$$L_5 = \{vw \mid v \in L,\ w \notin L\}$$

If $L$ is regular, then so is $L_5$. **(4)**

## Group C

C.1 (a) Prove that any combinational logic circuit can be designed using only XOR and AND gates. **(4)**

(b) Find a minimum product-of-sum expression of the Boolean function **(6)**

$$f(A,B,C,D,E) = \sum m(0,1,2,3,9,10,11,12,13,17,19,20,21,22,25,27,28,29) + \sum d(4,8,30).$$

No credits will be given for sum-of-product minimization.

C.2 A finite state machine takes a single-bit stream as input, and outputs a single-bit stream. The output of the circuit is 1 if the current input bit is the same as the previous input bit. The output is 0 otherwise. Here is an example.

Input: 100101110110...
Output: 001000110010...

(a) Draw the state diagram of the machine. Is it a Moore machine or a Mealy machine? **(3+1)**

(b) Design the finite state machine by a synchronous sequential circuit. The type of flip-flops that your circuit uses is your choice. **(6)**

C.3 If $\frac{z}{255} = y$, then $y = 256y - z$. This observation leads to a procedure for dividing a number $z$ by 255, using one subtraction to obtain each byte of the quotient. Draw a suitable architecture of the datapath for implementing this divide-by-255 algorithm without any multiplication or division operation. **(10)**

C.4 The following sequence of numbers represents memory addresses in a 64-word main memory: 0, 1, 2, 3, 4, 15, 14, 13, 12, 11, 10, 9, 0, 1, 2, 3, 4, 56, 28, 32. Classify each of the accesses as a cache hit, compulsory miss, capacity miss, or conflict miss, given the following cache parameters. Consider the following two configurations of cache memories while classifying the above accesses into cache hits and misses.

(a) Direct-mapped, 4-word lines, 4-line capacity **(5)**

(b) Two-way set-associative, 2-word lines, 4-set capacity, LRU replacement **(5)**

C.5 (a) Consider the following solution for the critical section problem. The two processes, $P_0$ and $P_1$, share the following variables:

```
boolean flag[2];
int turn;             /* initialized to 0 or 1*/
```

The structure of the Process $P_i$ ($i = 0$ or 1) is as follows:

```
do {
    flag[i] = true;
    while (flag[j]) {
        if (turn == j) {
            flag[i] = false;
            while (turn == j);
            flag[i] = true;
        }
    }
    Critical_section();
    turn = j;
    flag[i] = false;
    NonCritical_section();
} while(1);
```

"The above solution satisfies all the necessary requirements of the critical section problem." Prove or disprove with proper justification. **(3)**

(b) Consider a virtual memory system implementing two-level paged page table with 32-bit virtual address, 1KB page size, and 256 GB RAM. Each page table entry also stores a valid bit, a dirty bit, and two bits for read/write protection. What is the number of entries in the outer level page table? **(3)**

(c) Consider $N$ processes sharing the CPU in a round-robin fashion ($N \geq 2$). Assume that each context switch takes $S$ msec and that each time quantum is $Q$ msec. For simplicity, assume that processes never block on any event and simply switch between the CPU and the ready queue. Find the maximum value of $Q$ (as a functions of $N$, $S$ and $T$) such that no process will ever wait in the ready queue more than $T$ msec. **(2)**

(d) A system uses three page frames for storing process pages in main memory. It uses the Least Recently Used (LRU) page-replacement policy. Assume that all the page frames are initially empty. What is the total number of page faults that will occur while processing the page reference string given below? **(2)**

4, 7, 6, 1, 7, 6, 1, 2, 7, 2

C.6 (a) Consider the following set of jobs. What would be the average turnaround time if a shortest-remaining-time-first scheduling policy is used? Show the Gantt chart and all calculations. **($2\frac{1}{2}$)**

| Process | Arrival Time | Duration |
|---|---|---|
| $A$ | 0 | 12 |
| $B$ | 3 | 7 |
| $C$ | 6 | 2 |
| $D$ | 8 | 5 |
| $E$ | 9 | 2 |
| $F$ | 12 | 12 |

(b) The following program consists of three concurrent processes and three binary semaphores. The semaphores are initialized as: `S0=1, S1=0, S2=0.`

**Process P0:**

```
while(1) {
    wait(S0);
    print '0';
    signal(S1);
    signal(S2);
}
```

**Process P1:**

```
wait(S1);
signal(S0);
```

**Process P2:**

```
wait(S2);
signal(S0);
```

At most how many times will process P0 print '0'? **($2\frac{1}{2}$)**

(c) A scheduling algorithm SHD assigns priority proportional to the waiting time of a process. Every process starts with priority zero (the lowest priority). The scheduler re-evaluates the process priorities every $T$ time units and decides the next process to schedule. Which classical scheduling algorithm is identical with SHD if the processes have no I/O operations and all processes arrive at time zero? Explain. **(2)**

(d) A machine has a 32-bit logical address space and 8KB page. The page table is entirely in hardware, with one 32-bit word per entry. Once a process starts, the page table is copied to the hardware from memory, at one word every 100 nsec. If each process runs for 100 msec (including the time to load the page table), what fraction of the CPU time is devoted in loading the page tables. **(3)**