PhD Qualifier Examination

Department of Computer Science and Engineering

Saurav

Date: 27-Mar-2015 **Time:** 4 hours **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) A is a one-dimensional array containing n distinct integers. Span(A) is defined as the difference between the maximum and the minimum elements in A. Write the time complexity to evaluate Span(A) by the most efficient algorithm.

(b) The following array contains all positive integers up to 13:

$a = 12$
$d = 11$
$b = 7$
$c = 6.$

13	8	a	b	c	d	5	6	4	1	2	10	9	3

13 13 8 8 a a b b c c d d 5

Write the values of a, b, c, d so that it is a max-heap.

(c) A linked list contains an unsorted list of n distinct integers. Write the worst-case time complexity of the best algorithm that can copy and store these integers in a one-dimensional array in a sorted manner.

(d) S and T are two binary trees, each with n nodes. It is known that the asymptotic height of S is $O(\log n)$ but that of T is not so. For the most efficient algorithm, what will be the time complexities to measure the *exact* heights of S and T?

(e) A1, A2, A3 are three different algorithms for solving a problem. Their respective time complexities are as follows:

A1: $T(n) = \sum_{i=1}^{n} \Theta(i)$.

A2: $T(n) = T(3n/10) + T(7n/10) + 2n$.

A3: $T(n) = 3T(n/4) + 12n + 2\log n$.

Which one is asymptotically the best and which one is the worst?

A.2 A and B are two unsorted one-dimensional arrays. A contains m distinct integers, and B contains n distinct integers. However, an element of A may be equal to some element of B. Suggest an algorithm to construct a one-dimensional array C such that C stores $A \cup B$ with all distinct elements. Explain the time complexity of your algorithm. **(7 + 3)**

A.3 A is a one-dimensional array with n positive and negative integers in total, which are mixed up. Suggest an algorithm to find the maximum sum over all possible sub-arrays of A. Explain its time complexity. **(7 + 3)**

A.4 (a) For a graph $G = (V, E)$, provide a $O(|V| + |E|)$ time algorithm to check whether G is bipartite. **(5)**

(b) Prove that every nonempty DAG has at least one vertex with zero in-degree. **(5)**

A.5 (a) Prove that any two strongly connected components of a directed graph are either equal or disjoint. **(5)**

(b) Analyze the running time of Prim's algorithm for computing an MST of a connected undirected graph using binary heaps and adjacency lists. The heap can store a set of vertices (and not edges). **(5)**

A.6 Stirling numbers $S(n,k)$ (of the second kind) are defined for integers $n, k \geqslant 0$ recursively as:

$$
\begin{aligned}
S(0,0) &= 1, \\
S(n,0) &= 0 \quad \text{if } n > 0, \\
S(n,k) &= 0 \quad \text{if } k > n, \\
S(n,k) &= kS(n-1,k) + S(n-1,k-1) \quad \text{for } 1 \leqslant k \leqslant n.
\end{aligned}
$$

Write an *efficient* C function that takes n and k as arguments, and returns the value of $S(n,k)$. What is the time complexity of your C function (in terms of n and k)? **(8 + 2)**

A.7 Let us consider a binary search tree in which each node contains five fields: an integer key value, two child pointers (left and right), and two other integers min and max. The integer min at a node stores the minimum of all the key values stored in the subtree rooted at that node. Likewise, the max value at a node stores the maximum key value stored in the subtree rooted at that node.

(a) Declare a C data type to store a node in this tree. **(3)**

(b) Write a C function that inserts a key value x in this tree. Notice that a binary search tree does not contain duplicate key values (in multiple nodes). This means that if the new key value x is already present in the tree, no change is made. **(7)**

A.8 In a treasure island, there is a rectangular $m \times n$ mesh of cells. The cells are numbered by pairs (i, j) with $0 \leqslant i \leqslant m-1$ and $0 \leqslant j \leqslant n-1$. The entrance to the mesh is at the cell $(0,0)$. The (i, j)-th cells contains treasure worth $t_{i,j}$ (Indian Rupees). Each pair of adjacent cells (that is, cells sharing a wall, not only a corner) has a door which may be locked or unlocked. Write a C program that, given the two-dimensional treasure matrix $(t_{i,j})_{\substack{0\leqslant i\leqslant m-1\\ 0\leqslant j\leqslant n-1}}$ and the information about which doors are locked and which are unlocked, computes the total amount of treasure that you can collect from the mesh (entering at cell $(0,0)$). **(10)**

Group B

B.1 (a) Use mathematical induction to prove the following statement for $n \geqslant 1$: **(3)**

$$\frac{1}{1\cdot 2}+\frac{1}{2\cdot 3}+\frac{1}{3\cdot 4}+\cdots+\frac{1}{n(n+1)}=1-\frac{1}{n+1}.$$

(b) Prove that there are infinitely many prime numbers. You may use proof by contradiction. **(3)**

(c) Let A and B be finite sets, and let $\mathscr{P}(S)$ denote the power set of a set S. Prove or disprove: **(4)**

(i) $\mathscr{P}(A \cup B) = \mathscr{P}(A) \cup \mathscr{P}(B)$.

(ii) $\mathscr{P}(A \cap B) = \mathscr{P}(A) \cap \mathscr{P}(B)$.

B.2 (a) Let $X = \{1, 2, 3, \ldots, 30\}$. Define a relation $\approx$ on X as $x \approx y$ if and only if $x^2 - y^2$ is divisible by 5.

(i) Prove that $\approx$ is an equivalence relation. **(4)**

(ii) How many equivalence classes are induced by this relation? **(3)**

(b) Let A and B be sets of a finite size n. Show that if a function $f : A \to B$ is onto (surjective), then f is one-to-one (injective). **(3)**

B.3 (a) We hash 5 keys into 5 memory locations one by one. Assume each key is independently and uniformly hashed into the memory locations.

(i) What is the probability that the first x records do not produce any collision? **(3)**

(ii) What is the expected value of x when the first collision occurs? **(3)**

(b) Suppose that Mike selects a ball by fist picking one of two boxes at random and then selecting a ball from this box at random. The first box contains 4 red balls and 3 blue balls, and the second box contains 2 red balls and 5 blue balls. What is the probability that Mike picked a ball from the second box if he has selected a red ball? **(4)**

B.4 Consider the language

$$L_4 = \{w \in \{a, b, c\}^* \mid \text{the first and the last symbols in } w \text{ are different}\}.$$

Notice that the empty string ε does not belong to L_4.

(a) Write a regular expression for L_4. **(4)**

(b) Design a DFA/NFA/ε-NFA to accept L_4. **(6)**

B.5 The following two sets are defined over the alphabet $\{a,b,c\}$. One of these sets is regular, and the other not. Which one is what? Give proper justifications. **(5 + 5)**

(a) $L_{5a} = \{w_1 c w_2 \mid w_1, w_2 \in \{a,b,c\}^*\}$.

(b) $L_{5b} = \{w_1 c w_2 \mid w_1, w_2 \in \{a,b,c\}^*,\ |w_1| = |w_2|\}$. (Here, $|w_i|$ is the length of the string w_i.)

B.6 (a) Give a context-free grammar for the following language: **(6)**

$$L_6 = \{w \in \{a,b\}^* \mid w \text{ is not a palindrome}\}.$$

(b) Show derivations of the strings *aababba* and *aabaaba* using your grammar of Part (a). **(4)**

Group C

C.1 (a) Using a 5-variable K-map, find the minimum SoP expression of **(5)**

$$f(a,b,c,d,e) = \sum m(0,1,6,10,12,14,16,17,26,30).$$

(b) Realize $f(a,b,c) = \sum m(2,4,5,7)$ with a 4-to-1 multiplexer module (you can use inverters too). **(5)**

C.2 A sequential machine with two input wires and a single output has a behavior specified in the following transition table:

Present State	Next state Input 00	01	10	11	Output Input 00	01	10	11
a	*c*	*b*	*a*	*a*	0	0	0	1
b	*d*	*b*	*b*	*e*	0	1	1	1
c	*e*	*d*	*e*	*a*	0	0	0	1
d	*b*	*d*	*b*	*a*	0	1	1	1
e	*a*	*d*	*c*	*c*	0	0	0	1

Demonstrate that this behavior could be exhibited by a machine with fewer states. Draw the state diagram for the reduced machine. **(10)**

C.3 (a) Consider the `URISC` instruction:

```
label: urisc dest,src1,target
```

which stands for the operation: subtract operand1 (dest) from operand2 (src1), replace operand2(src1) with the result, and jump to target address (target) in case of negative result, else go to the next instruction.

Using the above `URISC` instruction *only*, write the following instructions: **(2 + 3)**

(i) `uadd dest,src1,src2          # dest = (src1) + (src2)`

(ii) `uswap src1,src2             # exchange (src1) and (src2)`

(b) A designer implemented an 8-bit unsigned adder with an input from his boss that the sum is always less than 255. Hence he left out the hardware required to generate the carry out. But on the close-to-delivery date, his boss wanted the carry out signal. So, help the designer to design an efficient circuit to compute the final carry out. **(5)**

C.4 (a) A byte-addressable memory with 32 bits of address has to be assisted with a cache memory. The cache size is 64 KB, and each cache line has 16 bytes.

Show the various parts of the address and identify which parts of the address are used to access the following cache configurations: **(2 + 3)**

(i) A Direct Mapped Cache

(ii) A 2-Way Set Associative Cache

(b) Show by a neat diagram how 128K × 8 SRAM chips can be used to build a 256K × 32 memory unit. The SRAM chips have input ports i) Din (Input Data Lines), ii) Addr (Address Lines), iii) CS (Chip Select), and iv) Dout (Output Data Lines). The resultant memory unit should have a) Data In (Input Data Lines), b) Data Out (Output Data Lines), and c) Address (Input Address Lines). **(5)**

C.5 (a) Let `m[0]...m[4]` be mutexes (binary semaphores, initialized to 1) and `P[0]...P[4]` be processes. Suppose that each process `P[i]` executes the following:

```
wait (m[i]); wait (m[(i+1) mod 4]);
signal (m[i]); signal (m[(i+1)mod 4]);
```

Could this cause (i) Deadlock (ii) Starvation? Justify with logic. **(4)**

(b) Consider N processes sharing the CPU in a round-robin fashion ($N \geqslant 2$). Assume that each context switch takes S msec, and the 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. **(3)**

(c) Suppose that we have a program that consists of eight pages (numbered 1 through 8), and we have four frames of physical memory for it. Find the minimum number of page faults possible for the following page-reference string:

1 2 3 2 5 6 3 4 6 3 7 3 1 5 3 6 3 4 2 4 3 4 5 1

Estimate the number of page faults for the same page-reference string when LRU page-replacement scheme is applied to it, for two different numbers of available frames: 3 frames and 4 frames. **(3)**

C.6 (a) Consider the following solution to the critical-section problem. Two processes P_0 and P_1 share the variable:

```
boolean flag[2]       /* initialized to 0 */
```

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

```
do {
    flag[i]=true;
    while(flag[1-i]);
    Critical Section
    flag[i]=false;
    Remainder Section
} while(1);
```

Prove or disprove: The above solution satisfies *all* the necessary requirements of a solution to the critical-section problem. **(3)**

(b) Consider the two-dimensional array

```
int A[100][100];
```

where the element `A[0][0]` is stored at location 200, in a paged system with pages of size 200. A small process is in the virtual Page 0 (locations 0 to 199) for manipulating the matrix; thus, every instruction fetch will be from the virtual Page 0. Let there be three physical frames in the main memory. Assume that the first frame is occupied by the process, and the other two frames are initially empty. Compute the number of page faults if the process runs: (i) Program I, (ii) Program II. **(4)**

Program I

```
for (j = 0; j < 100; j++)
    for (i = 0; i < 100; i++)
        A[i][j] = 0;
```

Program II

```
for (i = 0; i < 100; i++)
    for (j = 0; j < 100; j++)
        A[i][j] = 0;
```

(c) What is Belady's anomaly? Justify with explanation why the LRU page-replacement algorithm is free from Belady's anomaly. **(1 + 2)**