PhD Comprehensive Examination

Department of Computer Science and Engineering

Date: 14-March-2013 **Time:** 4 Hours **Maximum Marks:** 100

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

Group A

A.1 Write only the answers of the following questions. No need to write any explanation.

(a) Write the asymptotic upper bound of $T(n) = T(3n/10) + T(7n/10) + 10n$. **(2)**

(b) The array representing a max-heap is

20	12	10	8	4	7	6	5	2	3

Write the value of the element stored in the parent node of the node containing the smallest element, i.e., '2'. **(2)**

(c) Given an unordered list L of n integers, what is a tight lower bound on the worst case time complexity to find a/the pair of elements from L, which have the minimum difference among all possible pairs in L? **(2)**

(d) What is a tight lower bound on the average-case time complexity for successful search in a complete binary tree (which is not necessarily a binary search tree). **(2)**

(e) An undirected graph G containing n nodes and $O(n \log n)$ edges is represented by an adjacency matrix. Write the time complexity that will be required by an optimal algorithm to prepare the adjacency list of G from its adjacency matrix. **(2)**

A.2 $G(V, E)$ is an undirected, unweighted graph, represented by an adjacency list. It is known that G is a forest comprising of some trees. Suggest an efficient algorithm to find a set of edges $F \subset (V \times V) - E$ so that $G'(V, E \cup F)$ is a single tree (i.e., find a set of edges from $(V \times V) - E$ to connect all the trees into a single tree). Explain its time complexity. **(7 + 3 = 10)**

A.3 T is a height-balanced binary search tree containing n distinct integers. Suggest an algorithm to report the elements of T lying in the interval $[a, b]$. Assume that $n >> b - a$.
Explain its time complexity as a function of n, a, b. **(7 + 3 = 10)**

A.4 Write a C function **int SubPal(char *S, int *start, int *len)** that takes a null-terminated character string S as a parameter, and checks if any substring of S is a palindrome (a substring is any set of consecutive characters in the string). The function returns 0 if no such substrings are found. It returns 1 if at least one such substring exists, in which case the starting index of the substring is returned in *start* and the length of the substring is returned in *len*. If more than one such palindrome substrings exists, the starting index and length of any one such substring is returned.

For example, if $S =$ "*character*", then the function returns 1 and sets $*start = 2$ and $*len = 3$ for the palindrome substring "*ara*". However, if $S =$ "*palindrome*", the function returns 0, and the values in $*start$ and $*len$ are irrelevent (can be anything).

(10)

A.5 You are to implement a hash table to store at most 1000 integers. The hash fuction used for any key k is $k\%1000$. The collision resolution scheme used is open addressing.

(a) Define a datatype HASHTABLE to represent a hash table to store at most 1000 integers.

(b) Write a C function **int Insert(HASHTABLE H, int k)** that inserts a key k into a hashtable H. The function returns 1 if the insert is successful, 0 if the table is full.

C.3 (a) Can a non-preemptive CPU scheduler be called when some I/O that a process is waiting for completes? Justify your answer clearly.

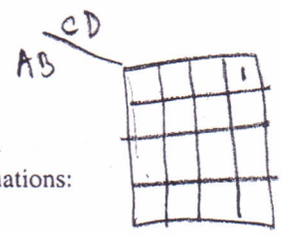
(b) "The page table of a process is kept in main memory. To access anything in main memory in a system with logical addressing, the translation of a logical address to a physical address is needed. Therefore, to access the page table itself, a logical address must be coverted to a physical address, which cannot be done without accessing the page table. Hence, paging cannot be implemented" - where is the flaw in this argument? Justify your answer clearly.

(c) Does a context switch have any effect on the TLB? Justify your answer.

(3 + 4 + 4)

C.4 (a) Implement the four Boolean functions listed using three half-adder circuits:

$$D = A \oplus B \oplus C$$
$$E = A'BC + AB'C$$
$$F = ABC' + (A' + B')C$$
$$G = ABC$$



(b) Design a sequential circuit with JK flip-flops to satisfy the following state equations:

$$A(t+1) = A'B'CD + A'B'C + ACD + AC'D'$$
$$B(t+1) = A'C + CD' + A'BC'$$

(5 + 5 = 10)

C.5 (a) You are an engineer at Outtel, a start-up company aspiring to compete with Intel via its new processor technology that outperforms the latest Intel processor by a factor of **2.5** on floating point instructions. To achieve this level of floating point performance, the design team had to make compromises that led to a 20% average increase in execution times of all other instructions. You are in charge of chosing benchmarks that would showcase Outtel's performance edge.

(i) What is the minimum required fraction f of time spent on floating point instructions in a program on the Intel processor to show a speed up of 2 for Outtel?

(i) If on the Intel processor, execution time of a floating point instruction is on an average 3 times as long as other instructions, what does the fraction in your answer to part (i) mean in terms of the instruction mix of the benchmark?

(b) Suppose a 2KB cache has set-associative address mapping. There are 16 sets, each containing four cache blocks or lines. The memory address size is 32 bits, and the smallest addressable unit is the byte. Label the components of the 32-bit address for accessing the cache memory.

(5 + 5 = 10)

C.6 Cache memories refer to a faster intermediate form of memory within a larger memory hierarchy. The following sequence of numbers represent 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, 15, 14, 13, 12, 0, 1, 2, 3. Classify each of the accesses as a cache hit, compulsory miss, capacity miss, or conflict miss, given the following cache parameters:

(i) Direct mapped, 4-word lines, 4-line capacity.

(ii) Two-way set associative, 4-word lines, 2-set capacity, LRU replacement.

(5 + 5 = 10)

Group B

B.1 (a) Prove by contrapostion : If n is a positive integer with $n^2 > 100$, then $n > 10$.

(b) Prove by induction: Every integer greater than 1 is a product of prime numbers.

(5 + 5 = 10)

B.2 (a) Show that composition of functions is associative.

(b) Prove that among any five points in the plane with integer coordinates, there are at least two points such that the center of the line segment that connects them also has integer coordinates.

(5 + 5 = 10)

B.3 (a) Show the recurrence relation for computing $M(n)$, where $M(n)$ is the number of moves needed to move n disks from peg i to peg j in the Tower of Hanoi problem.

(b) A test for a disease that affects 0.1% of the population is 99% effective on people with the disease (that is, it says they have it with probability 0.99). The test gives a false reading (saying that a person who does not have the disease is affected with it) for 2% of the population without the disease. What is the probability that someone selected at random and given a test for the disease will have a positive test? What is the probability that someone who has positive test results in fact has the disease?

(4 + 6 = 10)

B.4 (a) Let L_1 be the language over $\{0, 1\}$ consisting of all strings that have a 1 in the third symbol from the end (e.g., the string 0100 is in L_1). Draw a Deterministic Finite State Automata (DFA) containing minimum number of states that recognizes L_1.

(b) For a DFA A recognizing language L, interchanging final and non-final states gives an automaton that recognizes $\bar{L}$, the complement of L. Is this also true for Non-deterministic Finite State Automaton (NFA)? If your answer is yes, give a proof, if no, give a counterexample.

(c) Consider the following language L_2 defined over the alphabet $\{0, 1\}$.

L_2 = The set of all strings s such that the number of 0s and number of 1s in s differ by no more than 3.

Prove that L_2 is regular.

(4 + 2 + 4)

B.5 (a) Carefully state the Pumping Lemma for regular languages.

(b) Use the pumping lemma to prove that the following language L_3 over the alphabet $\{0, 1\}$ is not regular.

$$L_3 = \{0^n 1^n | n \geq 0\}$$

(3 + 7 = 10)

B.6 (a) Give a context free grammar for the following language L_4. The alphabet Σ is $\{0, 1, \#\}$.

L_4 = { w : w starts and ends with the same symbol, the length is odd, and the middle symbol is # }.

Draw the parse tree of the following string with the above grammar:

0101#1110

(b) Prove that if L is context free, then L^* must be context free.

(5 + 5 = 10)

Group C

C.1 (a) List clearly how a running process can be preempted and a new process run on a system using round-robin scheduling.

(b) An OS gives two system calls *sleep*() and *wakeup*(p). The *sleep*() call just puts the calling process (say x) to sleep until it is woken up by a *wakeup*(x) call made by some other process. The *wakeup*(x) call wakes up the specified process x if it is sleeping; if x is not sleeping when the *wakeup*(x) call is made, the call has no further effect. The following C–like pseudo–code is proposed for a naïve solution to the *Producer–Consumer Problem* on this OS:

```
#define N 100 // number of slots in the buffer
int count = 0; // number of items currently in the buffer

void producer (void)
{
  int item;
  while (TRUE) {
    item = produce_item();
    if (count == N) sleep();  // if buffer is full, sleep until woken up
    insert_item(item);
    count++;                  // increment number of items in buffer
    if (count == 1) wakeup(consumer);  // was buffer empty?
  }
}

void consumer (void)
{
  int item;
  while (TRUE) {
    if (!count) sleep();     // if buffer is empty, sleep
    item = remove_item();
    count--;
    if (count == N - 1) wakeup(producer); // was buffer full?
    consume_item(item);
  }
}
```

Unfortunately, this solution runs into trouble sometimes, with both the producer and the consumer going to sleep forever. Explain how this might happen in brief, clearly stating the possible sequence of operations that can cause this. Assume that no atomic operations are available.

(5 + 5)

C.2 (a) Suppose we have a program that consists of eight pages (numbered 1 through 8), and we have four page 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

(b) Estimate the number of page–faults for the following page reference string when the FIFO page–replacement scheme is applied to it, for two different number of available frames – 3 frames and 4 frames:

1 2 3 4 1 2 5 1 2 3 4 5

(5 + 5)