CSL 101 - Introduction to Computers and Programming
03 May 2007, 8:00 AM to 10:00 AM
Major Exam
Maximum Marks: 120

1. [5+5+5+5+5+5=30 Marks] Consider an extension of the binary search algorithm, where we divide the array A in which we are searching into 3 parts at each step, instead of 2 parts. The recursive search procedure continues in one of the selected ranges.

(a) Indicate the recurrence relation representing the computational complexity of this algorithm, if the array size is n.
(b) Solve the recurrence. What is the computational complexity of this algorithm?
(c) Is the complexity different from that of binary search? Explain clearly.
(d) What does the recurrence relation become if we divide the array into k parts in each step?
(e) Solve for the recurrence in step (d) above. What is the computational complexity of the algorithm in (d)?
(f) Verify your solution in (e) above by substituting $k = n$ and give an argument that the solution is reasonable.

2. [7.5+7.5+7.5+7.5=30 Marks] We are required to maintain a LINKED LIST L using one or more STACK data structures. That is, we are given the STACK operations: INIT, PUSH, POP, and EMPTY. Using only these, we should be able to perform the following operations on L:
(a) INIT (x): create a new empty LIST
(b) INSERT (x): insert a new element x into the LIST
(c) DELETE (x): delete the node containing x from the LIST
(d) SEARCH (x): search for element x; if found return TRUE, else return FALSE

Give algorithms for each of the above operations. Remember you are not allowed to create a LIST. You are only allowed to create a STACK, and realise the list operations using an equivalent STACK operation (or sequence of operations). Remember that the POP operation REMOVES the element from the top of the stack.

3. [5+5+5+5=20 Marks] What does the following program print? Explain why.

```
#include   <stdio.h>
int  a,d;
void  f  ()  {
  a++;  d++;
}
void  p  (int  r,  int  s)  {
  s = r;
  switch   (r)  {
    case  0:  f  ();
    case  5:  f  ();
    case  10:  f  ();
    default:    break;
  }
}
main  ()  {
  int  a,b,c;
  a = 0;  b = 0;  c = 5;  d = 6;
  p  (c,  b);
  printf   ("a  =  %d,  b  =  %d,  c  =  %d,  d  =  %d\n",   a,  b,  c,  d);
}
```

4. [20 Marks] Develop an algorithm REVERSE for determining the decimal digit reversal of a given integer. It takes as input a positive integer parameter x, and returns the integer that results when the decimal digits of x are reversed. For example:

REVERSE(12355) should return 55321
REVERSE(9) should return 9
REVERSE(1221) should return 1221
REVERSE of a negative number is not defined.

5. [10+10=20 Marks] Consider the assembly-language simulator you developed in Assignments 5,6,7. Instructions in a program are first loaded into memory, and the execution of each instruction involves:

- first accessing memory to fetch the instruction
- possibly reading some operands from memory, if required by the instruction
- possibly writing the results to memory, if required by the instruction

(a) Suppose a program executes m ADD instructions, n SUB instructions, p BRAG instructions, q STORE instructions, and one final HALT instruction. Compute the total number of memory accesses for the program. (Reading an operand from memory counts as one access. Writing the result to memory counts as one access).

(b) Suppose a total of k instructions have been executed in a program. What is the minimum number of memory accesses? What is the maximum number of memory accesses? Explain both answers.