

**CS 2604 Data Structures and File Management  
Midterm Exam**

**Summer II 2003  
22 July 2003**

Name: \_\_\_\_\_

Score \_\_\_\_\_

Pledge: On my honor, I have neither given nor received unauthorized aid for this exam.

Signed: \_\_\_\_\_

Instructions:

This exam is closed book and closed notes. Calculators or similar devices are not allowed nor are they necessary for this test. Write your answers on the spaces provided. For multiple choice questions, encircle the letter of the best answer. You may use the back sides of these sheets for your scratch work and notes—if you need extra sheets, the instructor will provide them to you.

I. Multiple Choice (20 points)

1. In C++, which of the following best corresponds to the notion of an Abstract Data Type (ADT)?
  - a. private members of a class
  - b. all the methods of a class
  - c. an abstract class with only pure virtual methods
  - d. a class with fully implemented methods
2. All programs are algorithms.
  - a. True
  - b. False
3. Which values for  $c$  and  $n_0$  will prove that  $3n^2$  is  $O(n^3)$ ?
  - a.  $c = 1, n_0 = 1$
  - b.  $c = 2, n_0 = 1$
  - c.  $c = 2, n_0 = 2$
  - d. None of the above
4. All *complete* binary trees with an odd number of nodes are also *full* binary trees.
  - a. True
  - b. False

5. A tree with at least two nodes has at least as many leaves as it has internal nodes.
  - a. True
  - b. False
  
6. Suppose a binary tree has only three nodes A, B, and C, and you are given that the post-order traversal for the tree is B-A-C. The pre-order traversal for the tree is:
  - a. C-A-B
  - b. A-B-C
  - c. C-B-A
  - d. None of the above
  - e. A definite pre-order traversal cannot be determined from the information given
  
7. Suppose you are given that a general tree has 4 nodes and has height 2. How many leaves does this tree contain?
  - a. 1
  - b. 2
  - c. 3
  - d. 4
  - e. The number of leaves cannot be determined from the information given.
  
8. An internal node in a Huffman tree correspond to:
  - a. a letter
  - b. a sum of frequencies of letters
  - c. the number of bits needed for a letter
  - d. none of the above
  
9. Which of the following sorting algorithms perform in  $O(n)$  time in the best case?
  - a. Insertion Sort
  - b. Bubble Sort
  - c. Selection Sort
  - d. All of the above
  - e. None of the above

10. Which of the following sorting algorithms perform in  $O(n^2)$  time in the worst case?

- a. Insertion Sort
- b. Bubble Sort
- c. Selection Sort
- d. All of the above
- e. None of the above

II. Running Time Complexity and Order Arithmetic (15 points: 10,5)

1. For the following C++ code fragment, find  $T(n)$ , the exact number of multiplications performed (operations to count are shown in bold font) in terms of  $n$  (there should be no summations in your final answer). Assume that  $n$  is a power of 2.

```
product = 1;
for( s = 1; s < n; s *= 2 )
    for( i = 1; i <= s; i++)
        product *= i;
```

$T(n) =$  \_\_\_\_\_

2. Encircle ALL statements that are correct:

$T(n)$  is  $\Theta(n^2)$

$T(n)$  is  $O(n)$

$T(n)$  is  $\Omega(\log n)$

III. Comparing List Implementations. (30 points: 5,5,15,10)

1. Assume that the size of a data element is three times the size of a pointer (factor = 3). If the size of the array in an array implementation of a list is  $n = 100$ , and the current number of elements in the list is  $c = 60$ , which list data structure is more space efficient, the singly-linked list implementation or the array implementation? Explain your answer.

---

---

---

---

2. Generalize your answer in #1. That is, when is an array more efficient than a singly-linked list (or vice versa)? Express your answer using an inequality condition in terms of factor,  $c$ , and  $n$ .

Array more efficient when: \_\_\_\_\_

Linked list more efficient when: \_\_\_\_\_

3. The following is the **setpos()** method described in class for a singly-linked list:

```
template <class Elem>
bool
LList<Elem>::setpos( int pos )
{
    if ( (pos < 0) || (pos > rightcnt+leftcnt) )
        return false;
    rightcnt = rightcnt + leftcnt - pos;
    leftcnt = pos;
    fence = head;
    for ( int i = 0; i < pos; i++ )
        fence = fence->next;
    return true;
}
```

Doubly linked-lists have a prev pointer for its nodes. It is possible to take advantage of this pointer (together with the tail member of LList<Elem>) to make setpos() slightly more efficient in some cases. Rewrite setpos() so that this advantage is realized.

4. Fill in the following table with either  $O(1)$ ,  $O(n)$ , or  $O(\log n)$ . These represent the costs of the methods for the given list implementation. Assume worst-case analysis.

Method	Array	Singly-Linked List	Double-Linked List
setpos()			
insert()			
append()			
prev()			

IV. Binary Trees (25 points: 5,10,5,5)

1. Draw an example of a binary tree with at least 6 nodes that is full but not complete.

2. Draw the binary tree that corresponds to the following sequential representation: AB//CD//GEF///H//

- The following array of integers exhibits a max-heap. Describe the contents of the array after the element 100 is removed from the heap. Note that the corresponding tree should remain complete

Before:

100	60	30	10	52	28	7	5	9	47	
-----	----	----	----	----	----	---	---	---	----	--

After:

--	--	--	--	--	--	--	--	--	--	--

- Suppose you wanted to insert the following elements in a binary search tree of integers: 3, 7, 12, 16, 50, 38, and 90. In what order should you insert these integers so that the resulting tree is complete? Assume that the BST is empty to begin with.

---

V. Project-related Questions (10 points: 5,5).

- For the first project, if you were implementing a chromosome using a doubly-linked list, how many next and prev pointers do you need to update when you are performing a translocation on:

- two adjacent blocks on a single chromosome: \_\_\_\_\_
- two blocks from different chromosomes: \_\_\_\_\_

You may assume that none of the blocks above include a beginning or ending element, and that no block is flipped (reversed) during the operation.

- For our current programming project, I allowed you to implement the time-ordered event queue using either a linked-list or a min-heap. Both implementations have pros and cons regarding the time-complexity of their operations. Describe briefly the advantage(s) and disadvantage(s) of these two implementations.

---



---



---



---