Name:________________________________

ECE 608: Computational Models and Methods, Fall 2005
Test #2
Thursday, November 10 2005

- Your exam should have 12 (twelve) pages.
- Pages 5, 7 11 and 12 are intentionally left blank.
- Write your name on this page and at least one other page.
- Closed book, closed notes.
- Switch off and put away cell-phones/pagers
- Apportion your time carefully.
- Numbers in brackets represent points for that question. Points add up to 100.
- Good luck.

Prob.	Max.	Score
I	10	
II	10	
III	20	
IV	20	
V	15	
VI	25	
Total	100	

Name:______________________________

I. (10 points) Categorize the following statements as **True** or **False**.

   a. Hashing using chaining is unstable as the load factor α approaches 1.

   b. Deletion in open address hashing is simpler if the probe sequences are stored in a doubly-linked list.

   c. The average complexity of an successful search in open address hash table with load factor α is given by $\frac{1}{\alpha}\ln\left(\frac{1}{1-\alpha}\right)$.

   d. Quadratic probing eliminates secondary but not primary clustering.

   e. Universal hashing guarantees that the probability of hashing two keys is no larger than 1/m (for an m-entry table).

II. (10 points) ‘n’ numbers are to be sorted. Each of the n numbers is represented as a 4*m bit binary number. Each number could be viewed as:

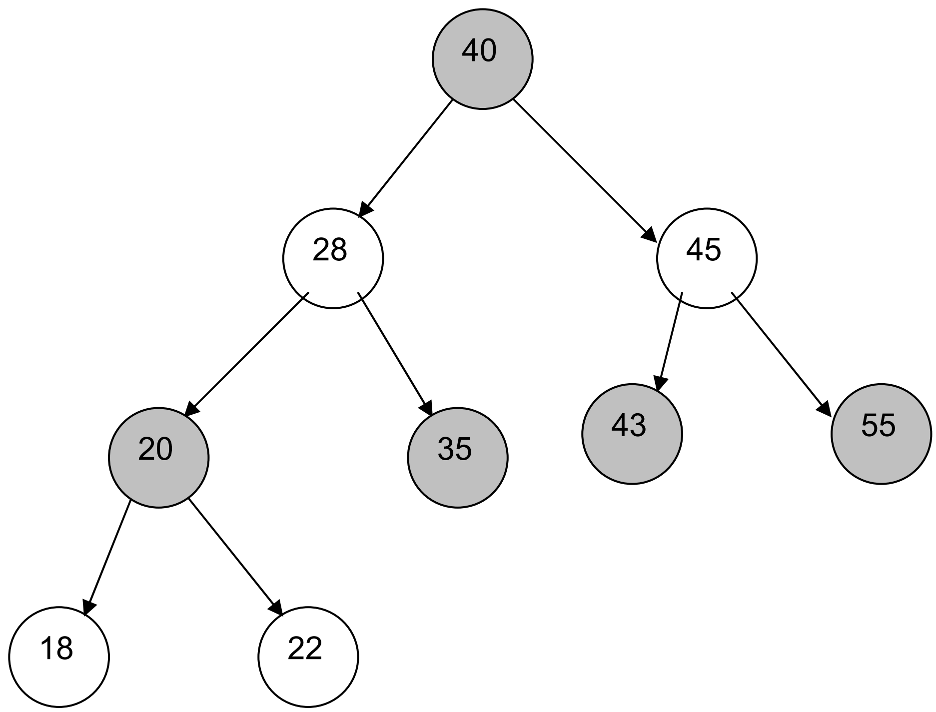
   a. an m-digit, hexadecimal (i.e., base 16) number OR
   b. a 2*m digit number to base 4.

What is the complexity of radix sort (using counting sort to sort individual digits) for the two different views?

III. (20 points)

a. (10 points) Consider a binary search tree in which each node contains (in addition to the key) the number of nodes in its left and right subtrees. When TREE_DELETE(T, z) is called on a binary search tree T and node z with two (non-NIL) child-nodes, there are two possible courses of action that result in correct deletion. What are these two possibilities? Provide an algorithm (pseudocode) to improve the balance of the tree by intelligently choosing between the two possibilities.

b. (10 points) Insert the key 16 into the red black tree below. Perform all required rotations and recoloring as necessary to restore properties of red-black trees. If you need multiple iterations of recoloring/rotation show all intermediate steps leading to the final RB-tree. (The sentinel node, nil[T], which acts as dummy leaf nodes as well as the parent of the root node is not shown to reduce clutter.)



IV. (20 points)
The Fibonacci sequence computation is an often overlooked example of dynamic programming.

A. (5 points) Provide pseudocode to describe a traditional top-down recursive computation of the ith Fibonacci number Fib(i).
B. (5 points) Argue that the problem satisfies all the requirements for a dynamic programming approach.
C. (5 points) Provide the pseudocode for a dynamic-programming based bottom-up computation of Fib(i). What is the time complexity of this version?
D. (5 points) Provide the pseudocode for a top-down memoized implementation for computing Fib(i). What is the time complexity of this version?

Name:______________________________

Name:______________________________

V. (15 points) Apply Huffman's encoding to determine the prefix codes for the symbols (and the frequencies) shown below.

Character	Frequency	Fixed length code	Huffman code
A	85	000	
B	71	001	
C	43	010	
D	73	011	
E	52	100	
F	60	101	
G	89	110	
H	51	111	

A. (10 points) What is the Huffman code for each character? Show all the work required to arrive at the final codes and fill in the codes in the table. (The prefix-code tree with frequencies at all intermediate nodes must be shown for full credit.)

B. (5 points) By what factor does your code compress the length of encoding the phrase ‘ABBGHFFFEFDCGC’ compared to a fixed length encoding of the same phrase. (If the code does not result in a compression or if it results in an expansion, your answer should state it explicitly.)

VI. (25 points) A bridge has to be built over a river that is W feet wide. The challenge is to minimize the cost of building the bridge.

- The engineers have decided that the bridge needs n columns (C_1, C_2, … , C_n) to support the bridge. The cost of buiding the bridge is the sum of the costs of building each support column.
- The columns are supposed to be built at specified distances (in feet) from the left riverbank specified by the vector $< w_1, w_2, \ldots, w_n>$.
- For ease of notation, we refer to the left bank as C_0 with w_0=0 and the right bank as Cn+1 with w_{n+1} = W. Assume that C_0 and Cn+1 are prebuilt with zero cost.
- The columns have to be built one after another due to manpower constraints.
- The cost of building a column is proportional to the distance between the previously built columns (including riverbanks) that are to the immediate left and right of the given column.
- For example, the cost of constructing the first column (say C_k) is always $C_{n+1} – C_0$ = W.
- The cost of constructing the second column (say C_j) is
  - o if j > k, $w_{n+1} – w_k$ = W-w_k
  - o if j < k, $w_k – w_0$ = w_k

Use dynamic programming to describe an algorithm to compute the optimal cost of building the bridge. Your algorithm should also provide the optimal solution (i.e., the order in which the columns must be built to achieve the optimal cost.)

For full credit, your description must explicitly state (a) the recursive way to compute the optimal cost, (b) the base case values for the recursion and (c) the pseudocode for computing the optimal solution and (d) the complexity of the algorithm.