Full Name: ____________________

Andrew login ID: ____________________

CS 15-451, Spring 2001

Midterm Exam

March 6, 2001

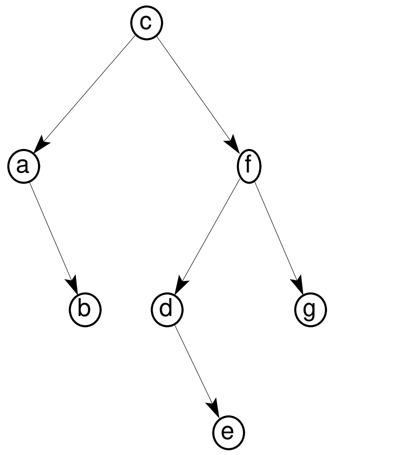
Instructions:

- Make sure that your exam is not missing any sheets, then write your full name and Andrew login ID on the front.
- Write your answers in the space provided below the problem. If you make a mess, clearly indicate your final answer.
- The exam has a maximum score of 100 points.
- The problems are of varying difficulty. The point value of each problem is indicated. Pile up the easy points quickly and then come back to the harder problems.
- Answer all questions on this form. If necessary, use the back of the page (and put a pointer on the front).
- This exam is CLOSED BOOK. You may only use one page of notes. Good luck!

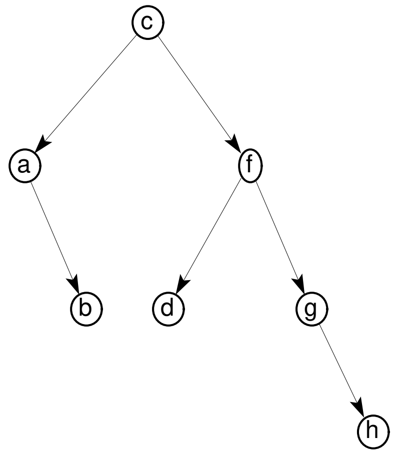
1 (15):
2 (15):
3 (20):
4 (25):
5 (25):
TOTAL (100):

Problem 1. Trees (15 points):

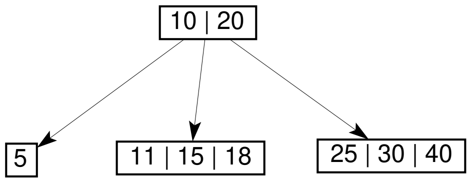
A. Show the result of the *first step* in splaying node e in the following tree.



B. Show the result of the *first step* splaying node h in the following tree.



C. Draw a picture of the following B-tree ($t = 2$) after inserting items 12 and 50.



Problem 2. Recurrences (15 points):

In each of the questions below, check exactly one box.

A. The median algorithm based on splitting the list into groups of 3 leads to the recurrence $t(n) = t(n/3) + t(2n/3) + n$ for the running time. Which function below solves this recurrence asymptotically?

n	
$n \lg n$	
n^3	
none of the above	

B. If we use a binary trie to sort n numbers of size k bits, what is the running time?

$\Theta(k\,n)$	
$\Theta(n \lg k)$	
$\Theta(k\,n^2)$	
none of the above	

C. Suppose a divide-and-conquer algorithm has non-recursive overhead $n\sqrt{n}$, leading to the recurrence $t(n) = 2\,t(n/2) + n\sqrt{n}$ for the running time. Which function below solves this recurrence asymptotically?

n	
$n \lg n$	
$n\sqrt{n}$	
none of the above	

Problem 3. Matrix Multiplication (20 points):
Suppose we have matrices $A_1, A_2, \ldots, A_{n-1}, A_n$ of sizes $p_0, p_1, \ldots, p_n$, so that A_i has dimension $p_{i-1} \times p_i$. We have seen in class that the total number of scalar multiplications needed to compute the product of these matrices depends strongly on the evaluation order (i.e., the places where we put parentheses).
Suppose instead that we wish to minimize the number of entries in the largest matrix generated when evaluating the product. For example, suppose we multiply matrices A_1, A_2, and A_3 of dimensions 2×10, 10×5, and 5×20, respectively. Then the multiplication order $(A_1 \times A_2) \times A_3$ first creates a 2×5 (10 entry) matrix and then a 2×20 (40 entries). The multiplication order $A_1 \times (A_2 \times A_3)$ creates a 10×20 (200 entry) matrix and then a 2×20 (40 entries). We would therefore say that the "cost" of the first order is 40, while the second is 200.

A. Describe a dynamic programming solution to the problem of finding an evaluation order that minimizes the cost function defined as the total number of entries in the largest matrix generated while computing the product. As usual, you may confine yourself to showing how to compute the optimal cost (rather than having to determine the actual order).

B. Analyze the running time of your algorithm in terms of n, the number of given matrices.

Problem 4. Hashing (25 points):
Suppose we hash n objects into a hash table with m buckets using a hash function that distributes objects uniformly into the buckets. Define α to be the table load, i.e., $\alpha = n/m$.
In the following problems, you may find the following identity useful:

$$\lim_{x \to \infty} \left(1 + \frac{a}{x}\right)^x = e^a. \tag{1}$$

A. Write a formula in terms of k, m, and n for the probability that a given bucket i will have exactly k objects hashed to it.

B. Write a formula in terms of m and α for the expected *fraction* of buckets that are empty.

C. Using Equation 1, give a formula in terms of α for the expected fraction of the buckets that are empty for large values of m.

D. Write a formula in terms of m and α for the expected fraction of buckets that contain exactly one item.

E. Using Equation 1, write a formula in terms of α for the expected fraction of the buckets that contain exactly one item, for large values of m.

Problem 5. Amortized Analysis (25 points):
Suppose we want to implement a *table* data structure, where a table is like an array, except that it can grow dynamically. In particular, the operation Append(A, x) increases the length of A by 1 and inserts element x in this new position.

In Section 18.4 of CLR, they describe an implementation that doubles the amount of memory allocated for a table every time more space is required. Thus, the sequence of allocation sizes will be powers of two: an allocation of length 1 gets doubled to 2, which gets doubled to 4, etc.

The drawback of this scheme is that almost 1/2 of the allocated space could be wasted. For example, a table of 33 elements would require a memory allocation large enough to hold 64 elements. 31 of these locations would be wasted.

One way to reduce the wasted space would be have the memory manager allocate blocks whose sizes are perfect squares. At any time, a table of size n would be stored in a block of size b^2, where $b = \lceil \sqrt{n} \rceil$. If adding an element to this table would yield $n + 1 > b^2$, then we would get a block of size $(b + 1)^2$ from the allocator, copy the n elements over to the new block, and add the new element. The "cost" of this Append would be n. The cost of appending an element where there is already space would be just 1.

A. Prove that the amount of wasted space for a table of size n would be $O(\sqrt{n})$.

B. Prove that for any sequence of n Append operations starting from an empty table, the amortized cost of any Append operation would be $O(\sqrt{n})$.

(*blank page*)