ICS 261 Fall 2000
Data Structures
Midterm 2—Version A
November 21, 2000

This is a closed-book test. You may not use calculators, books, or notes during the exam.

Please write all of your answers on the answer sheet, and write your name and ID number, as well as the test version, on both sides of the answer sheet. This is version A. You may keep the exam.

I suggest you look over the entire midterm before starting.

The maximum possible score is 40 on Part I and 60 on Part II, for a total of 100 points.

Part I. Written answers

Point values for each question in this part are given in square brackets after the question number. The notation "explanation optional" on some problems means that a completely correct final answer will get full credit, though an explanation could conceivably help you get partial credit.

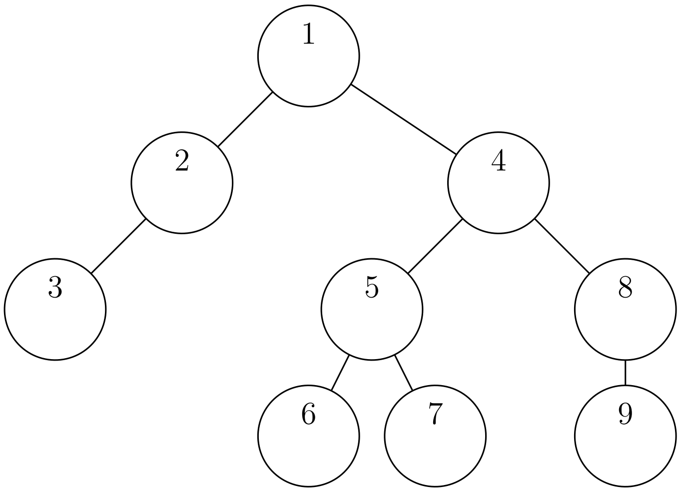
1. [10—Explanation optional] Suppose that we have a binomial heap on 41 nodes.

   a) How many separate binomial trees does the heap have, and what are their sizes?

   b) How many nodes in the heap are children of the roots of these trees? (Hint: you don't need to draw the whole tree to answer this.)

2. [10] When performing the lowest common ancestors algorithm, we know that there may be many nodes in T that ϕ maps into the same node b in B. Suppose that T has 50 nodes. What is the largest possible number of nodes in T that can map to the same node in b? In other words, what is the largest possible value of $|\phi^{-1}(b)|$ for a node b in B? Fully justify your answer.

3. [10—Explanation optional] Below is a tree with its nodes numbered in preorder. Suppose we wish to perform the LCA algorithm on this tree. Give the value of INLABEL for each node in the tree, and give the value of ASCENDANT for just the nodes whose preorder numbers are 7 and 9. (Write the INLABEL numbers below the preorder numbers on the tree on the answer sheet, and separately write, as 4-bit binary numbers, the two requested ASCENDANT numbers.)



4. [10] Suppose we are given three strings $a = a_1a_2\ldots a_n$, $b = b_1b_2\ldots b_n$, and $c = c_1c_2\ldots c_n$. Assume that code to produce a suffix tree in linear time is given. Show how to find the largest string x such that x is a substring of b and c, but not a substring of a, in linear time. Explain your answer briefly but clearly.

Part II. Multiple choice.

For each question, choose the *best* answer and write its letter clearly as a large plain capital on the answer form. Each question is worth 5 points, and there is no penalty for guessing.

It's a good idea to read each question carefully and look at all the choices. (Sometimes they may be in a different order than you would expect.)

5. Which of the functions shown below grows most slowly as n approaches infinity?

   A. $n3^n(\log n)^3$ B. $n^2 3^n$ C. $n^5 2^n$ D. n^{10} E. $(\log n)^{100}$

6. Let T be a binary tree with root r. Suppose that the first node in the preorder traversal of T is also the first node in the inorder traversal of T. What can you conclude? (Give the strongest valid conclusion.)

   A. T has an even number of nodes.
   B. The right subtree of r is empty.
   C. The left subtree of r is empty.
   D. Both the left and right subtrees r are empty.

7. What is the last digit of the minimum possible number of keys in an AVL tree of height 7? (Assume that we measure height in the extended tree. Recall that the number of keys is just the number of internal nodes in this tree.)

   A. 0 or 5 B. 1 or 6 C. 2 or 7 D. 3 or 8 E. 4 or 9

8. For which of the following types of trees is the worst-case time for an individual insertion $\Theta(n)$?

   A. Binary search trees
   B. AVL trees
   C. Red-black trees
   D. Splay trees
   E. A and D

9. Suppose that we define a hypothetical data structure consisting of a single integer i in the range $-n$ to n, with n even, on which four operations are possible: Increment, Decrement, JumpUp, and JumpDown.

   Increment increases the value of i by 1 and Decrement decreases the value of i by 1, each at a cost of one unit; however, if either operation causes the value of $|i|$ to become equal to $n/2$, then the value of i is reset to 0 at an additional cost of $n/2$ units.

   JumpUp increases the value of i by n and JumpDown decreases the value of i by n; either operation costs just one unit.

   Assume that operations are never performed that would move i outside the range $-n$ to n. Initially i is 0. Which potential function below could be used to show that the amortized cost of each of the four operations is $O(1)$ (with a constant independent of n)?

   A. $|i|$
   B. $|n - i|$
   C. $n/2 - \big|n/2 - |i|\big|$
   D. i^2
   E. $n - i^2$

10. Suppose that we have an application for heaps in which the number of DecreaseKey operations will be much greater than the total number of the other operations. In addition, we only care about the total time to perform all of the operations, not the time required by individual operations. Which type of heap is most efficient for this application?

   A. d-heaps
   B. Binomial heaps
   C. Fibonacci heaps

11. What is maximum possible number of keys in a red-black tree of black height 2? (Measure height in the extended tree; remember that the internal nodes each contain one key but the external nodes do not have keys. Also, recall that the external nodes are always colored black.)

A. 15 B. 8 C. 7 D. 3 E. 2

12. Let I_m denote the average internal path length of a binary search tree, assuming that we build the tree by starting with an empty tree and then inserting m distinct keys with all permutations being equally likely. Now suppose that for some odd number n, we

a) make a list of all $n!$ permutations of the integers $1, 2, \ldots, n$,

b) cross out all permutations in which the median element (i.e., $(n+1)/2$) does not come first, and then

c) for each of the remaining permutations, build a binary search tree by starting with the empty tree and inserting the elements in that order.

What will be the average internal path length of the trees we construct?

A. $n - 1 + I_{\lfloor n/2 \rfloor}$

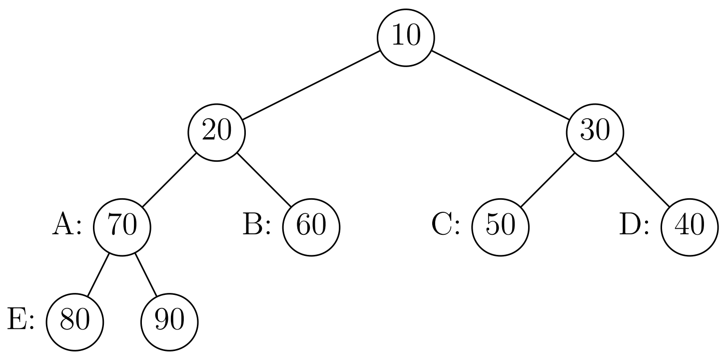
B. $n - 1 + 2I_{\lfloor n/2 \rfloor}$

C. $n - 1 + I^2_{\lfloor n/2 \rfloor}$

D. I_{n-1}

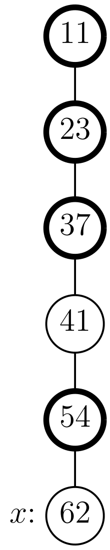
E. $(I_n + I_1)/2$

13. Below is a 2-heap (i.e., a d-heap with $d = 2$). If we delete the minimum where will the key 90 appear in the resulting heap?



A. A B. B C. C D. D E. E

14. Suppose that the path from the root to some node x in a Fibonacci heap is as shown below; marked nodes are indicated by bold circles. Then we perform a DecreaseKey to lower the value of the key in x to 15. When the operation has completed, how many new roots will appear in the top-level tree list?



A. 1 B. 2 C. 3 D. 4 E. 5

15. When we do a DeleteMin from a Fibonacci heap, we remove the node containing the minimum value, add the children of the deleted node to the list of roots, and then perform a consolidate operation which involves linking roots of equal degree until all roots remaining have distinct degrees. Suppose that just before the consolidate operation there are 8 roots in the list, with degrees 1,1,2,4,4,5,6,6. After the consolidate, how many roots will there be?

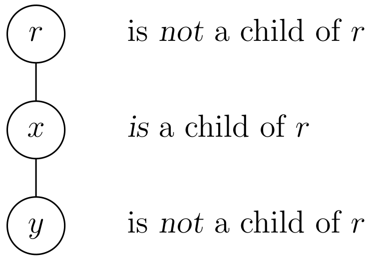
A. 2 B. 3 C. 4 D. 5 E. 6

16. Suppose we are working with a Union-Find data structure as described in class. Node z is at a distance of 4 from the root r of its tree. If we now perform a Find on z, by how much will the degree of r change?

A. It will decrease by 1.
B. It will increase by 1.
C. It will increase by 2.
D. It will increase by 3.
E. It will increase by 4.

Appendix. Notes added in class when the test was given

1. a) "size" means number of nodes in each tree.
   b) The final answer is one number—the total number of children of all the roots of the trees in part (a). (You can explain how you got the answer if you want.)



11. Will be discarded.