ICS 261 Fall 1999
Data Structures
Final Exam—Version A
4:00–6:00 pm, Thursday December 9, 1999

Directions: This is a multiple-choice exam. For each question, choose the *best* answer and clearly mark the corresponding choice on the answer form. There is no penalty for guessing.

On the front of the answer form, please

a) write your name,

b) put your ID number in the last 8 squares of the space for ID numbers, leaving the first two blank, and then bubble in the appropriate boxes, and

c) for question 21, and for "Test Form," please be sure to mark choice A so we know that you had Version A of the exam. Just leave "Exam #" blank. You need not fill out anything on the back of the form.

Questions 1–20 are each worth the same amount.

You may not use calculators, books, or notes during the exam. (No problem involves long messy computations if you approach it right.) You may not look toward other people's answers or talk with anyone other than the instructor during the exam.

Your exam should have 22 questions. 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.) Also, just because a question looks similar to one you may have seen on a previous exam, do not assume that it is identical; there could be some changes.

You may keep the exam.

1. Using the formal definition of O-, Ω-, and Θ-notation, which of the choices below are true?

   A. $n^2 - n \in O(n^3)$

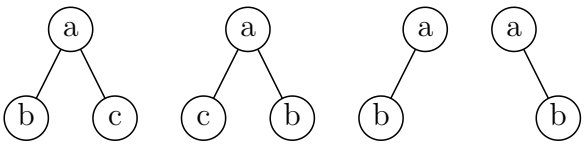
   B. $n^2 - n \in \Omega(n^3)$

   C. $n^2 - n \in \Theta(n^3)$

   D. A and B

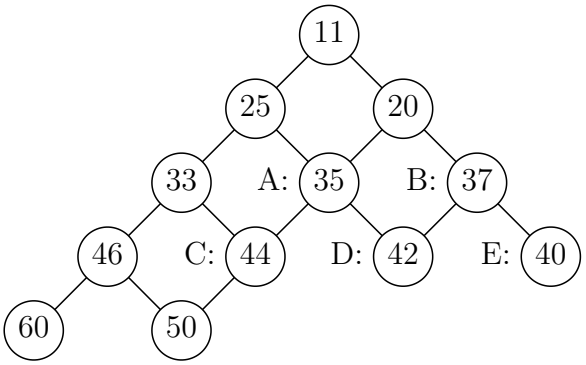
   E. A and C

2. Suppose that we are thinking of the trees below as rooted (not ordered) trees. How many distinct trees are there? (As usual, the top node is the root.)



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

3. Below is a beap. Keys are shown inside each node, and some of the nodes have been given letters (to their left) so that we can refer to them. If we delete the key 11 in the usual way, where will key 50 be after the deletion (including the operations to make the structure a valid beap again) is complete?



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

4. What does it mean to say that an algorithm is on-line?

   A. It works correctly even if some of the lines in the data have the same x- or y-coordinates.

   B. It gives the results of operations as they are performed, without needing to see the entire sequence of operations in advance.

   C. It works correctly if the data consists entirely of lines, but is not guaranteed to work correctly if some circular arcs are included in the input.

   D. It runs in $O(1)$ time per operation.

5. Below is a list which we are maintaining using the Transpose (TR) update rule. If we access the 97, in which location will the 48 be after the operation? (The locations are given as numbers above each box. Note that we are not using the Move-to-front (MF) rule.)

1	2	3	4	5	6	7	8	9
22	55	48	73	97	11	34	61	82

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

6. Suppose we have a B-tree of order m. (This is equivalent to what we would have called an $(\lceil m/2 \rceil, m)$-tree.) Which inequality below best characterizes the number k of keys that can appear in a node which is neither a leaf nor the root?

   A. $\lceil m/2 \rceil \leq k \leq m$

   B. $\lceil m/2 \rceil + 1 \leq k \leq m + 1$

   C. $\lceil m/2 \rceil - 1 \leq k \leq m - 1$

   D. $\lceil m/2 \rceil + 1 \leq k \leq m - 1$

7. Let $[a, b]$ and $[x, y]$ be two intervals. Which condition below is *equivalent* to $[a, b] \cap [x, y] \neq \emptyset$? (Don't worry if you don't remember it; just figure which choice must be correct.)
   A. $x \geq b$ and $y \geq a$
   B. $x \geq a$ and $y \leq b$
   C. $x \leq b$ and $y \geq a$
   D. $x \leq a$ and $y \leq b$

8. When we discussed the planar point location problem, we divided the plane into slabs which had none of the endpoints in their interior; we maintained a tree for each slab. How were we able to do all of the preprocessing in $O(n \log n)$ time?
   A. Scanning the plain from left to right, we added and deleted points in persistent trees.
   B. We processed the endpoints in batches of size $\Theta(\log n)$.
   C. We arranged the slabs into groups of size $\Theta(\log n)$, and then, for each group, processed only the queries involving that group.
   D. We approximated the regions in the plane by ellipsoids.

9. We mentioned a possible problem when performing deletions in hash tables, that can cause one to fail to find keys on subsequent searches. Which collision resolution method is susceptible to this problem?
   A. (Separate) chaining.
   B. Linear probing.
   C. Double hashing.
   D. A and B.
   E. B and C.

10. What did universal hash functions, for a table size m, enable us to achieve?
    A. Guaranteeing that a new key could be inserted into the hash table in worst-case $O(1)$ time.
    B. Producing a collision resolution strategy that was invulnerable to deletions.
    C. Producing a bound of $1/m$ on the probability that two keys K and K' would hash to the same location, whenever $K \neq K'$.
    D. Producing a hash function that worked for any type of key: integer, floating point, or string.

11. What is a possible advantage of open addressing over (separate) chaining?
    A. It eliminates the need for extra storage for pointers.
    B. It can work well even when the load factor (i.e., the ratio of the number of keys stored to the number of values of the hash function) is about 2 or 3.
    C. It can function correctly even in the presence of duplicate keys.
    D. It makes it possible to store n keys using a total of $o(n)$ space.

12. What is a possible advantage of (separate) chaining over open addressing?

    A. It eliminates the need for extra storage for pointers.

    B. It can work well even when the load factor (i.e., the ratio of the number of keys stored to the number of values of the hash function) is about 2 or 3.

    C. It can function correctly even in the presence of duplicate keys.

    D. It makes it possible to store n keys using a total of $o(n)$ space.

13. Given a tree T built of nodes containing Left, Right, and Data fields, and the path to a node x in T at a distance of d from the root of T, it is possible to create the new tree T' resulting from a splay at x in $O(d)$ time while creating only $O(d)$ new nodes, without changing the original tree T. (This is done by using ideas similar to those we described for persistent binary trees.) Suppose that we use path copying as in ordinary binary search trees followed by this type of splaying to produce a persistent version of splay trees. How much space would we use to maintain all n of the trees produced by performing n operations on an initially empty tree?

    A. $O(n)$ B. $O(n \log n)$ C. $O(n \log^2 n)$ D. $O(n^2)$

14. Which of the choices below gives us the number of distinct tree shapes possible for a binary tree on n nodes?

    A. The prime numbers

    B. The modular numbers

    C. The Catalan numbers

    D. The Martinique numbers

15. Let a_i and b_i, for $1 \leq i \leq n$, be real values such that each b_i is nonnegative and

$$\sum_{i=1}^{n} b_i = 1,$$

    and let $f(x)$ be a convex function (for example, $f(x) = x^2$). What does Jensen's inequality enable us to conclude?

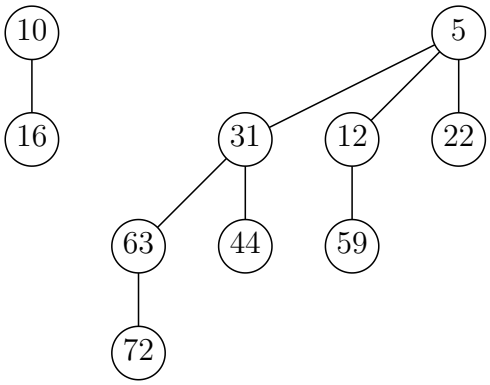
    A. $\sum_{i=1}^{n} b_i f(a_i) \leq f\left(\sum_{i=1}^{n} b_i a_i\right)$

    B. $\sum_{i=1}^{n} a_i f(b_i) \leq f\left(\sum_{i=1}^{n} a_i b_i\right)$

    C. $\sum_{i=1}^{n} b_i f(a_i) \geq f\left(\sum_{i=1}^{n} b_i a_i\right)$

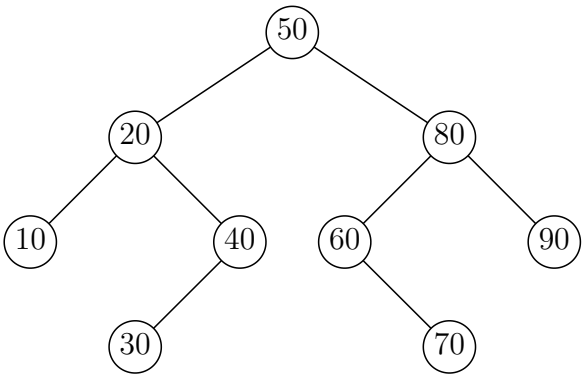
    D. $\sum_{i=1}^{n} a_i f(b_i) \geq f\left(\sum_{i=1}^{n} a_i b_i\right)$

16. Below is a binomial heap on 10 keys. Note that it consists of two trees. If we delete the minimum (and make the result back into a binomial heap as usual), how many trees will be in the heap?



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

17. Below is a binary search tree. If we delete the 50 using the algorithm we discussed, what value will be in the root of the remaining tree? (Because of symmetry, there are two natural ways to do it. If you don't remember which of these we used, that's OK: only one of the choices below is possible if you try both ways.)



A. 60 B. 70 C. 80 D. 90

18. For which of the following were we *not* guaranteed that $O(1)$ rotations sufficed to restore the balance condition?

A. Deletion in red-black trees

B. Insertion in red-black trees

C. Deletion in AVL trees

D. Insertion in AVL trees

19. When we analyzed Fibonacci heaps we used a potential function $\Phi(h) = a + 2b$, where h was a Fibonacci heap, a was the number of trees in h, and b was the number of marked non-root nodes in h. What was the key observation in the proof that DecreaseKey used $O(1)$ amortized time?

    A. Each cascading cut step increased a by 1 and decreased b by 1.

    B. Each cascading cut step decreased a by 1 and increased b by 1.

    C. Each consolidate step increased a by 1 and decreased b by 1.

    D. Each consolidate step decreased a by 1 and increased b by 1.

20. (This is based on a generalization of a problem that was used on Midterm 1 this quarter.) Suppose we define a hypothetical data structure consisting of two integers i and j in the range 1 to n, with n being even, on which just four operations are possible: Up increments j, Down decrements j, Right increments i, and Left decrements i. Initially $i = j = n/2$. Each operation costs 1 unit, except that if either i or j reaches 0 or n, *both* i and j are reset to $n/2$ at a total cost (covering both resets) of $n/2$ time units.

    We wish to use a potential function argument to show that the amortized cost of each operation is at most 2. Which of the potential functions below would be suitable for the argument?

    A. $|i - n/2| \cdot |j - n/2|$

    B. $|i - n/2| + |j - n/2|$

    C. $|i + j - n|$

    D. $i + j - n$

    E. B and C

21. Mark A for this question, so we know that you had Version A of the midterm. Please also make sure that you marked choice A in the "Test Form" box.

22. I plan to let you know your grades by email. Is that OK with you?

    A. Yes

    B. No

    If the registrar does not have your correct email address, or you would like me to send the results to a different address, please write the address clearly above your name on the answer sheet. Alternatively, you can just put a note above your name saying you would like to pick up the results yourself at my office.