

**Instructor: Dr. Sartaj Sahni
Spring, 2003**

**Advanced Data Structures
(COP 5536 /AD 711R)
Exam 2 (Make-up)**

**CLOSED BOOK
75 Minutes**

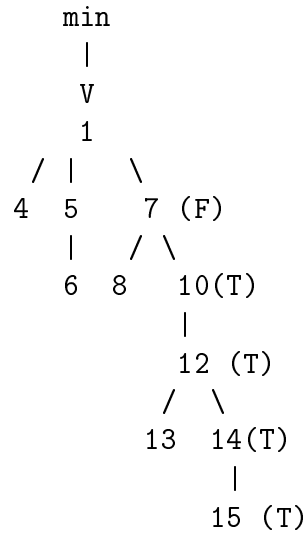
Name: _____

NOTE:

1. **For all problems, use only the algorithms discussed in class/text.**
2. All answers will be graded on correctness, efficiency, clarity, elegance and other normal criteria that determine quality.
3. The points assigned to each question are provided in parentheses.

1. (14)

- (a) (6) For the following min Fibonacci heap, assume that the *ChildCut* field of each node is TRUE except node 7. And the ChildCut of a root node is undefined.



Perform a *DecreaseKey* operation by changing 14 to 9. On the resulting Fibonacci heap, clearly label *ChildCut* values (Draw the resulting *Fibonacci* heap.)

- (b) (8) Show that if we start with an empty Fibonacci heap and do not perform cascading cuts, then it is possible for a sequence of Fibonacci heap operations to result in degree- k min trees that have only $k+1$ nodes, $k \geq 1$.

2. (8) Start with an empty *two-pass min* pairing heap,
 - (a) (4) Insert the following sequence of keys: 5, 8, 4, 12, 3, 14, 20, 15, and 9 in this order. Show the pairing heap after each insert.
 - (b) (4) Perform a *RemoveMin* operation on the resulting min heap of (a), showing each step.

3. (10) Recall that *inserting* a node into an *AVL* tree may require LL, LR, RL, or RR rotations. Draw AVL trees in which inserting a node requires an *RL* rotation. Remember that there are *three* cases for *RL* rotations. For each case, indicate a node to be inserted, perform an insert operation, and draw the AVL tree following the insertion.

4. (8) Draw a 2-3 tree with 11 elements (keys from 1 to 11) and height 3, where all nodes at levels 2 and 3 are 2-nodes (the root is at level 1). *Delete* the element in the rightmost node at level 2 and draw the resulting 2-3 tree. From the resulting tree, *delete* the *min* element. Draw the new 2-3 tree.

5. (10) For red-black trees,

(a) (7) Construct a red-black tree by inserting the keys in the following sequence into an initially empty red-black tree : 4, 9, 1, 13, 11, 7, and 5.

Use the *bottom – up* algorithm. Show each step.

(b) (3) For the resulting red-black tree of (a), perform the *Delete* operation for key value 13, showing each step.