Instructor: Dr. Sartaj Sahni
Summer, 2005

Advanced Data Structures
(AD 711R)
Exam 02

CLOSED BOOK
80 Minutes

Name: _____

PLEASE READ THE FOLLOWING INSTRUCTIONS CAREFULLY

1. **For all problems, use only the algorithms discussed in class/text.**
2. **Write your name at the top of every exam sheet.**
3. **Write your answers directly on the exam question sheet.** You may use scrap paper (supplied by your proctor) for work, but these will not be graded.
4. All answers will be graded on correctness, efficiency, clarity, elegance and other normal criteria that determine quality.
5. The points assigned to each question are provided in parentheses.
6. You may use only a pen or a pencil. No calculators allowed.
7. Do not write on the reverse side of the exam sheet.
8. Do not write close to the margins since those areas do not always make it through when faxed.

Name:

1. (11) For the following min Fibonacci heap. (The ChildCut field is shown in parentheses; ChildCut is undefined for the root.)

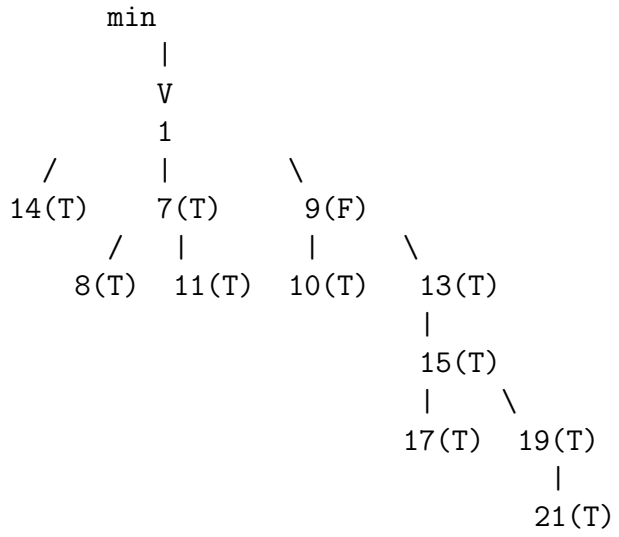


Figure 1. Min Fibonacci heap

(a) (5) For the min Fibonacci heap of figure 1, perform a *DecreaseKey* operation by changing 19 to 2. Draw the resulting *min Fibonacci* heap, clearly label *ChildCut* values.

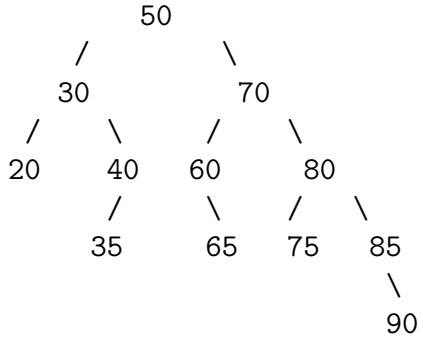
(b) (6) Perform a *DeleteMin* operation on the resulting Fibonacci heap of (a) , clearly label *ChildCut* values.

Name:

Continue work here if necessary.

Name:

2. (11) Consider the following AVL tree.



(a) (5) Delete key 40, show each step and specify each rotation type.

(b) (6) Start with the original AVL tree(i.e., the tree before the deletion key 40) and insert 87 and 91 in this order. Show each step and specify each rotation type.

Name:

Continue work here if necessary.

Name:

3. (12) For 2-3 trees,

   (a) (6) Insert the following sequence of keys: 9,8,7,2,6,1,4 and 3 into an initially empty 2-3 tree. Show each step.

   (b) (6) Delete the minimum key of the root node. Show each step.

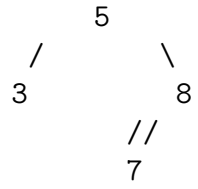
Name:

Continue work here if necessary.

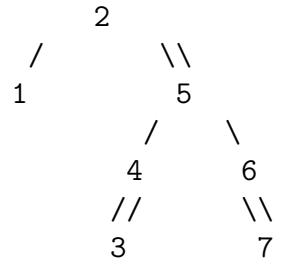
Name:

4. (16) For *red-black* trees, use the *bottom-up* algorithm for this problem. Double lines indicate a red edge and single line a black edge.

   (a) (8) Insert 6,10,1 and 2 in sequence into the following red-black tree.



   (b) (8) Consider the red-black tree below. Perform *Split(3)* operation, showing each step.



Name:

Continue work here if necessary.