Instructor: Dr. Sartaj Sahni
Spring, 2002

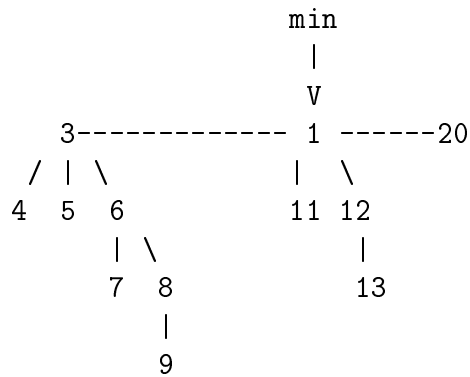Advanced Data Structures
(COP 5536 /AD 711R)
Exam 2

CLOSED BOOK
60 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.
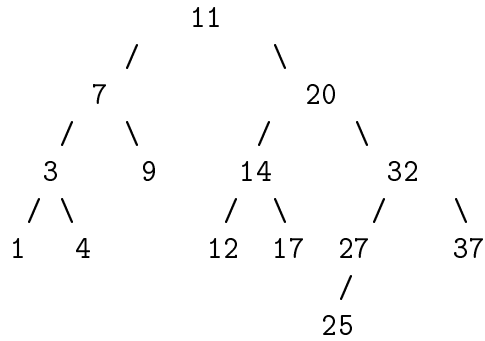3. The points assigned to each question are provided in parentheses.

1. (10) For the following three-tree min Fibonacci heap, assume that the *ChildCut* field of each node is TRUE.

```
                          min
                           |
                           V
         3------------- 1 ------20
        / | \           | \
       4  5  6         11 12
            | \           |
            7  8         13
               |
               9
```

(a) (4) Perform a *DecreaseKey* operation by changing 8 to 2. (Draw the resulting *Fibonacci* heap.)

(b) (6) Perform a *DeleteMin* operation on the resulting Fibobacci heap, clearly labelling ChildCut value. (Show each step)

2. (8) Start with an empty *two-pass* max pairing heap.

   (a) (4) *Insert* the following sequence of keys: 4, 5, 9, 8, 20, 12, 3, 14, and 15 in this order. Show the pairing heap after each insert.

   (b) (4) Perform a *RemoveMax* operation on the resulting heap of (a), showing each step.

3. (10) Consider the following AVL tree,

```
                   11
               /        \
             7            20
           /   \        /     \
         3       9    14        32
        / \          /  \      /    \
       1    4      12   17   27      37
                              /
                            25
```
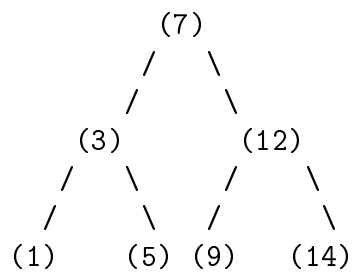
(a) (4) Delete key 9, show each step and specify each rotation type.

(b) (6) Start with the original AVL tree(i.e., the tree before the deletion of key 9) and insert 26 and 29, in this order. Show each step and specify each rotation type.

4. (10)For 2-3 trees,

   (a) (5) Insert 13 and 5 in sequence into the following 2-3 tree, showing each step.

```
              ( 4, 8)
          /      |       \
         /       |        \
     (1, 3)   (6, 7)     (10)
```
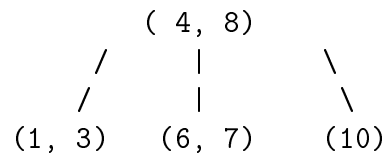
    (b) (5) Delete 7 and 14 in sequence from the following 2-3 tree, showing each step.

```
                (7)
               /   \
              /     \
           (3)        (12)
          /   \      /    \
         /     \    /      \
       (1)     (5) (9)    (14)
```

5. (12) For red-black trees,

    (a) (6) Insert 6, 10, 1, and 2 in sequence into the following red-black tree (double lines indicate a red edge and single line a black one) using the *bottom-up* (2-pass) algorithm. Show each step.

```
        5
       / \
      3   8
         //
          7
```
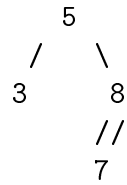
    (b) (6) For the resulting red-black tree of (a), perform the *split* operation for key value 6, showing each step.