Instructor: Dr. Sartaj Sahni
Spring, 2005

Advanced Data Structures
(COP5536)
Exam 01 - Make-up

CLOSED BOOK
50 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 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. (10) Consider two operations $put(x)$ and $get(n)$ on an initially empty jar. $put(x)$ pushes a ball x into the jar and $get(n)$ pulls out n balls from the jar (If the number of balls b in the jar is less than n, just pull out b balls). A $put(x)$ operation takes $O(1)$ time and a $get(n)$ operation takes *O(min{the number of balls in the jar, n})* time.

   Show that the amortized complexity of the *put* and *get* operations is $O(1)$.

Name:

Continue work here if necessary.

Name: ______________________________

2. (10) You are given 8 runs with 100, 200, 300, 400, 500, 600, 700, and 800 equal-length records. The block size is 100 records. The runs are to be merged using either an optimal 4-way or 8-way merge scheme. Assume that each merge is done using a loser tree.

   Determine the number of comparisons and the number of disk I/Os for both merge schemes. Which scheme do you recommend when all input, output, and CPU processing are sequential?

Name:

Continue work here if necessary.

Name: ____

3. (10) For the interval heap,

```
                5,95
              /      \
         15,75         25,85
         /   \         /   \
    35,65  25,45  35,75  81,82
```

   (a) (4) *Insert 90* into the interval tree, showing steps (Use the algorithm discussed in class).

   (b) (6) Perform *RemoveMin* from the original interval heap above, showing each step (Use the algorithm discussed in class).

Name:

Continue work here if necessary.

Name:

4. (10) Consider a height-biased min leftist tree:

   (a) (4) Convert the following min tree to a height-biased min leftist tree and label each node x with its shortest(x) value. Do this by swapping left and right subtrees as needed.

```
        3
      /   \
     4     5
          / \
         9   8
          \ / \
         15 13 21
```

   (b) (6) Draw the min leftist tree that results from when the *combine* operation is performed on the two min leftist trees. Show each step.

```
        2                      3
      /    \                 /    \
     4       6              7       5
   /   \    /   \          / \     /
  8     9  14    13       10  9  12
       /
      17
```

Name:

Continue work here if necessary.

Name: ______________________________

5. (10) For the min bionomial heap,

   (a) (4) *Insert* the keys in sequence: 6, 2, 5, 13, 10, 8, 3, 9, 1, 12, and 4 into an initially empty min bionomial heap (Use the algorithm discussed in class). Show the resulting min bionomial heap.

   (b) (6) Perform *RemoveMin* on the tree of (a) (Use the algorithm discussed in class), showing each step.

Name:

Continue work here if necessary.