Instructor: Dr. Sartaj Sahni
Fall, 2001

Advanced Data Structures
(COP 5536 /NTU AD 711R)
Exam 1

CLOSED BOOK
60 Minutes
Take one Week after Lecture 13

Name: ______________________________

SSN: ______________________________

Site Number: ______________________________

NOTE: All answers will be graded on correctness, efficiency, clarity, elegance and other normal criteria that determine quality. The points assigned to each question are provided in parentheses.

1. (10) Suppose that a sequence of n operations is performed on a data structure. The kth operation has a cost of $2\sqrt{k}$ whenever k is a perfect square (k = 1, 4, 9, 16, 25, etc), and otherwise it has a cost of 1. Use any one of the following methods of analysis to determine the amortized cost per operation: aggregate, accounting, or potential method. Please specify which method you are using. (noice that, find the smallest integer amortized cost)

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?

3. (10) given the interval heap shown below, show the final interval heap produced by the following sequence of operations: *insert*(20), *deleteMin*().

   (Use the algorithms discussed in class and showing steps)

```
                 (2,30)
             /               \
         (3,14)              (4,19)
      /         \           /      \
   (4,12)       (3,11)  (5,15)    (6,16)
   /    \        /    \
(4,10) (5,11) (5,9)  (4,7)
```

4. *(10)* Use the method discussed in class to combine the following two min-left trees into a single min-leftist tree. Show the final min-leftist tree as well as intermediate stages of the combine operation. Consider a height-biased min leftist tree.

```
         2                          7
        / \                        / \
       /   \                      /   \
      5     20                  12      15
     /\     /                   /       /
    /  \   /                   /       /
   11  17 24                 16      18
  /                          /\     /
 /                          /  \   /
31                         22  28 19
```

5. (10) For the min bionomial heap,

   (a) (5) *Insert* the keys in sequence: 13, 4, 7, 2, 11, 8, 6, 9, 1, 10, 5 and 3 into an initally empty min bionomial heap. (Use the algorithm discussed in class). Show the resulting min bionomial heap.

   (b) (5) Do *DeleteMin* on the tree of (a)(Use the algorithm discussed in class), showing each step.