Instructor: Dr. Sartaj Sahni
Summer 2012

Advanced Data Structures
(COP 5536)
Exam 1

CLOSED BOOK
60 Minutes

Name: _____

UFID: _____

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. (15 points) Assume that a linked list has $Add(x)$ and $DeleteOdds()$ operations, defined as:

   - $Add(x)$: Adds the element x to the end of the list
   - $DeleteOdds()$: Delete the first, third, fifth, etc., elements of the list

   The actual cost of $Add(x)$ is 2 and the actual cost of $DeleteOdds()$ is the number of elements in the list.

   What is the smallest integer amortized cost for the $Add()$ and $DeleteOdds()$ operations **respectively**? Show how you arrive at this.

Continue work here if necessary.

2. (15 points) 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.

   (a) 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.(7 points)

   (b) Assume that the time of I/O is 100 times of comparison time, which scheme is better when all input, output and CPU processing are sequential? (3 points)
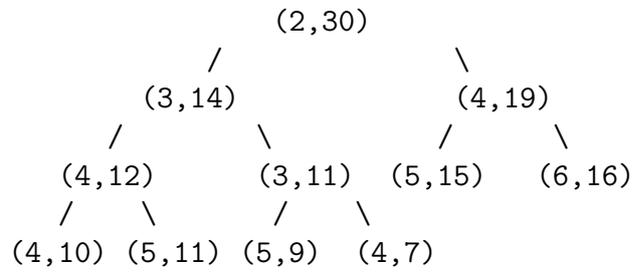
Continue work here if necessary.

3. (10 points) 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)
```

Continue work here if necessary.

4. *(10 points)* Perform the following operation sequence on an initially empty *min binomial heap* (showing each step).

Insert(6), Insert(2), Insert(4), Insert(23),Insert(10),Insert(20), RemoveMin, RemoveMin, Insert(3), Insert(8),Insert(25),RemoveMin.

(use the enhanced *RemoveMin* that combines trees of equal degree)

Continue work here if necessary.