Instructor: Dr. Sartaj Sahni
Summer, 2005

Advanced Data Structures
(COP5536)
Exam 01

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. (15) 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 1 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? Show how you arrive at this.

Name:

Continue work here if necessary.

Name:

2. (15) You are given 8 runs with 100, 200, 300, 400, 400, 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.

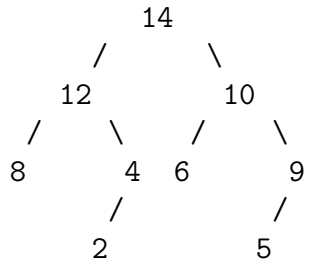
   Determine the number of comparisons and the number of disk I/Os for both merge schemes. Which scheme do you recommend when all input, output, and CPU processing are sequential?

Name:

Continue work here if necessary.

Name:

3. *(10)* Consider the following height-biased max leftist trees. Show the modified tree under each of the following operations. (Note: The two operations (a) and (b) are independent. Each of them starts from the given tree.)



(a) (4) Perform a $put(7)$. Label each node of the resulting tree with its s-value($shortest(x)$).

(b) (6) Perform a $RemoveMax()$ operation on the given tree above. Show each step.

Name:

Continue work here if necessary.

Name:

4. *(10)* 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)

Name:

Continue work here if necessary.