Instructor: Dr. Sartaj Sahni
Summer, 2001

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

CLOSED BOOK
50 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) 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 $O(1)$.

2. (10) You are given 5 runs with 10, 20, 20, 40, and 60 equal-length records to be merged into one, using a loser tree. It takes 2 seconds to read or write one block from/to disk and it takes 1 second to merge one block of records.

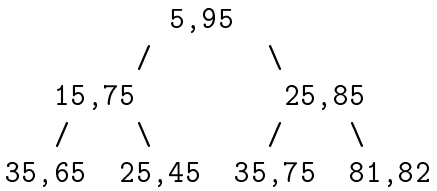
   Assume that all input, output, and CPU processing is sequential. The block size is 10 records.

   (a) (4) Give an optimal 4-way merging scheme.

   (b) (3) What is the total number of comparisons?

   (c) (3) Compute the total time taken by the optimal scheme.

3. (10) For the interval heap,

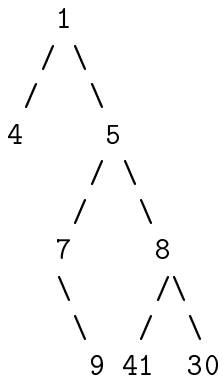


(a) (4) *Insert 90* into the interval tree, showing steps (Use the algorithm discussed in class).

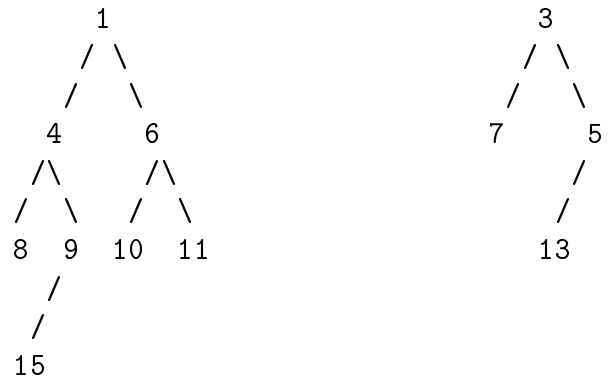
(b) (6) Perform *DeleteMin* from the original interval heap above, showing each step (Use the algorithm discussed in class).

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

   (a) *(5)* 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.



   (b) *(5)* Draw the min leftist tree that results from when the *combine* operation is performed on the two min leftist trees. (Following the algorithm in the text)



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