COP 5536
Advanced Data Structures

Spring 2019

Exam 1
CLOSED BOOK
60 Minutes

PLEASE READ THE FOLLOWING INSTRUCTIONS CAREFULLY

1. For all problems, use only the algorithms discussed in class.

2. Write your answers directly on the exam question sheet. You may use scrap paper (supplied by your proctor) for work, but these will not be graded.

3. All answers will be graded on correctness, efficiency, clarity, elegance and other normal criteria that determine quality.

4. You may use only a pen or a pencil. No calculators allowed.

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.

Last Name: _____
First Name: _____
UFID: _____

Q. 1 (14)	Q. 2 (12)	Q. 3 (12)	Q. 4 (12)	Total (50)

Question.1 (14):

Recall that a *QUEUE* is a first-in-first-out (FIFO) data structure with two basic operations ENQUEUE and DEQUEUE, and a *STACK* is a first-in-last-out (FILO) data structure with two operations PUSH and POP. Suppose that the actual cost of each PUSH and POP is 1. Following is an algorithm to implement a *QUEUE* with two *STACK*s:

Name the two STACKs as Stack1 and Stack2, we can implement the QUEUE as follows:
• ENQUEUE(x): PUSH x into Stack1
• DEQUEUE(x): If Stack2 is not empty, then simply POP from Stack2 and return the element. If Stack2 is empty, POP all the elements of Stack1, PUSH them into Stack2, then POP from Stack2 and return the result.

a) (5) What is the worst-case cost of the DEQUEUE and ENQUEUE operations as a function of queue size? Explain.

b) (9) Use any of the three methods (1) aggregate, (2) accounting, or (3) potential function, to find a constant integer amortized complexity for the ENQUEUE and DEQUEUE operations.

Continue work here if necessary.

Question 2 (12):

You are given 6 runs with 400, 500, 600, 800, 800, and 900 equal-length records to be merged into one, using loser tree. It takes 10 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 100 records.
(a) (6) 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.

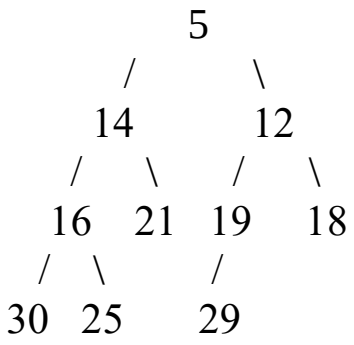
Continue work here if necessary.

Question 3 (12):

For the following height-biased min leftist tree,

(a) (6) Use the method discussed in class to **perform *Insert(10)***. Show the final min-leftist tree as well as intermediate stages of the operation.

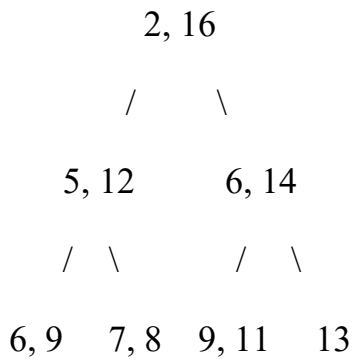
(b) (6) Use the method discussed in class to **perform *DeleteMin*** from the **original** tree. Show the final min-leftist tree as well as intermediate stages of the operation.



Continue work here if necessary.

Question 4 (12):

For the following interval heap,
(a) (6) Do ***Insert(19)***, showing each step.
(b) (6) Do ***DeleteMin*** from the resulting interval heap of (a), showing each step.



Continue work here if necessary.