Laboratory 11: Cover Sheet

Name: Saharath Kleips Date: 11/04/2014

Section: 1001

Place a check mark in the *Assigned* column next to the exercises your instructor has assigned to you. Attach this cover sheet to the front of the packet of materials you submit following the laboratory.

Activities	**Assigned:** Check or list exercise numbers	**Completed**
Implementation Testing	✓	
Programming Exercise 1	✓	
Programming Exercise 2		
Programming Exercise 3	✓	
Analysis Exercise 1	✓	
Analysis Exercise 2	✓	
	Total	

Laboratory 11: Implementation Testing

Name: Saharath Kleips Date: 11/04/2014

Section: 1001

Check with your instructor whether you are to complete this exercise prior to your lab period or during lab.

Test Plan 11-1 (Heap ADT operations)			
Test case	**Commands**	**Expected result**	**Checked**
+3+2+1---	Insert, Remove	Inserted 3 Removed 3 Empty List	

Laboratory 11: Programming Exercise 1

Name: Saharath Kleips Date: 11/04/2014

Section: 1001

Test Plan 11-2 (Priority Queue simulation results)		
Time (minutes)	**Longest wait for any low priority (0) task**	**Longest wait for any high priority (1) task**
10	1	4
30	2	8
60	2	8

Question 1: Is your priority queue task scheduler unfair—that is, given two tasks T_1 and T_2 of the same priority, where task T_1 is enqueued at time N and task T_2 is enqueued at time $N + i$ ($i > 0$), is task T_2 ever dequeued before task T_1?

The compare condition is if it is not equal, so there is no switch if T1 and T2 are equal. Therefor the queue is fair.

Question 2: If so, how can you eliminate this problem and make your task scheduler fair?

Laboratory 11: Programming Exercise 2

Name: Saharath Kleips Date: 11/04/2014

Section: 1001

Test Plan 11-3 (heapSort operation)			
Test case	**Array**	**Expected result**	**Checked**

Laboratory 11: Programming Exercise 3

Name: Saharath Kleips Date: 11/04/2014

Section: 1001

Test Plan 11-4 (The writeLevels operation)			
Test case	**Commands**	**Expected result**	**Checked**
+3+2+1w	Add Items, Print	3 21	

Laboratory 11: Analysis Exercise 1

Name: Saharath Kleips Date: 11/04/2014

Section: 1001

You can use a heap—or a priority queue (Programming Exercise 1)—to implement both a first-in, first-out (FIFO) queue and a stack. The trick is to use the order in which data items arrive as the basis for determining the data items' priority values.

Part A

How would you assign priority values to data items to produce a FIFO queue?

You would use a minimum heap and set priorities by when they arrive at the heap.

Part B

How would you assign priority values to data items to produce a stack?

You would use a maximum heap and set priorities by when they arrive at the heap.

Laboratory 11: Analysis Exercise 2

Name: Saharath Kleips Date: 11/04/2014

Section: 1001

Part A

Given a heap containing ten data items with distinct priorities, where in the heap can the data item with the next-to-highest priority be located? Give examples to illustrate your answer.

The next priority item will be the highest of the two children. After that, the highest priority item will be either of the siblings or one of its children.

```
        2
3 <
        1
or
        1
3<
        2
```

Part B

Given the same heap as in Part A, where in the heap can the data item with the lowest priority be located? Give examples to illustrate your answer. .

It will be located in the last two levels, the position however depends on when it is inserted into the heap.

```
        0
9<              6
        8<
                7
```

Turns into when 5 is added:

```
        5\
                0
9<
                6
        8<
                7
```