Assignment 3 – Data Structures and Algorithms
Deadline: Wednesday March 27 by 11:59 pm
Type: Individual Assignment
Weight: 8%

Theory Questions (20 points):

Q1 (5)

2.4.4 Is an array that is sorted in decreasing order a max-oriented heap. Why?

Q2 (10)

Draw the min-heap that results from the bottom-up heap construction algorithm on the following list of values: 10, 17, 15, 25, 40, 19, 45, 16, 12, 8, 18, 14, 13, 9, 20, 11, 13 Starting from the bottom layer, use the values from left to right as specified above. Show immediate steps and the final tree representing the min-heap. Afterwards perform the operation removeMin, 3 times and show the resulting min-heap after each step.

Q3 (5)

Suppose that your application will have a huge number of insert operations, but only a few remove the maximum operations. Which priority-queue implementation do you think would be most effective: heap, unordered array, or ordered array? Explain your answer.

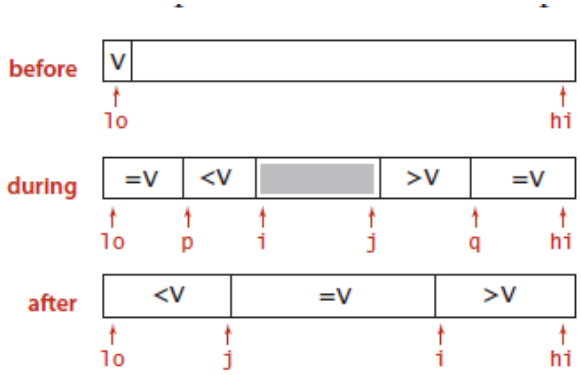
Programming Questions (80 points):

Q1 (10)

2.3.12 Show, in the style of the trace given with the code, how the entropy-optimal sort (3-way quick sort) first partitions the array B A B A B A B A C A D A B R A.

Q2 (15)

Fast 3-way partitioning. (J. Bentley and D. Mcllroy) Implement an entropy-optimal sort based on keeping item's with equal keys at both the left and right ends of the subarray. Maintain indices *p* and *q* such that *a[lo..p-1]* and *a[q+1..hi]* are all equal to a[lo], an index *i* such that *a[p..i-1]* are all less than *a[lo]*, and an index *j* such that *a[j+1..q]* are all greater than *a[lo]*. Add to the inner partitioning loop code to swap *a[i]* with *a[p]* (and increment p) if it is equal to *v* and to swap *a[j]* with *a[q]* (and decrement q) if it is equal to *v* before the usual comparisons of *a[i]* and *a[j]* with *v*. After the partitioning loop has terminated, add code to swap the items with equal keys into position.
Note : This code complements the code given in thetext, in the sense that it does extra swaps for keys equal to the partitioning item's key, while the code in the text does extra swaps for keys that are *not* equal to the partitioning item's key



Bentley-McIlroy 3-way partitioning

Q3 (15)

2.4.29 *Min/max priority queue*. Design a data type that supports the following operations: *insert*, *delete the maximum*, and *delete the minimum* (all in logarithmic time); and *find the maximum* and *find the minimum* (both in constant time). Hint Use two heaps.

Q4 (15)

2.5.9 Develop a data type that allows you to write a client that can sort a file such as the one shown below. (You need to create a file with the format of the input to test your program.)

Input (DJI volumes for each day)

```
 1-Oct-28       3500000
 2-Oct-28       3850000
 3-Oct-28       4060000
 4-Oct-28       4330000
 5-Oct-28       4360000
...
30-Dec-99     554680000
31-Dec-99     374049984
 3-Jan-00     931800000
 4-Jan-00    1009000000
 5-Jan-00    1085500032
...
```

output

```
19-Aug-40 130000
26-Aug-40 160000
24-Jul-40 200000
10-Aug-42 210000
23-Jun-42 210000
...
23-Jul-02 2441019904
17-Jul-02 2566500096
15-Jul-02 2574799872
19-Jul-02 2654099968
24-Jul-02 2775559936
```

Q5 (10)

3.1.1 Write a client that creates a symbol table mapping letter grades to numerical scores, as in the table below, then reads from standard input a list of letter grades and computes and prints the GPA (the average of the numbers corresponding to the grades).

A+	A	A–	B+	B	B–	C+	C	C–	D	F
4.33	4.00	3.67	3.33	3.00	2.67	2.33	2.00	1.67	1.00	0.00

Q6 (15)

3.1.3 Develop a symbol-table implementation `OrderedSequentialSearchST` that uses an ordered linked list as the underlying data structure to implement our ordered symbol-table APL