COMPUTER SCIENCE 21A (SUMMER, 2017)
DATA STRUCTURES AND ALGORITHMS

PROBLEM SET 4

Due Friday, Aug 4

- Your assignment should be submitted via Latte as a PDF by the date it is due.
- Name your PDF file LastnameFirstname-PS4.pdf
- Type each problem (i.e., Q1, Q2, …) on a separate page. Indicate the problem number on each page.
- If a problem requires pseudocode, each line of your algorithm should be numbered (see pseudocode in this assignment).
- Late submissions will not receive credit.

Q1. Given input {4371, 1323, 6173, 4199, 4344, 9679, 1989} and a hash function $h(x) = x \ mod \ 10$, show the resulting:

a. Separate chaining hash table
b. Hash table using linear probing
c. Hash table using quadratic probing
d. Hash table with second hash function $h_2(x) = 7 - (x \ mod \ 7)$

Finally, show the result of rehashing the hash table.

Q2. Let T be a heap storing n keys. Give an efficient algorithm for reporting all the keys in T that are smaller than or equal to a given query key x (which is not necessarily in T). Note that the keys do not need to be reported in sorted order.

Q3. Show how to implement the standard queue ADT using only a priority queue and one additional integer instance variable.

Q4. A min-max heap is a data structure that supports both *deleteMin* and *deleteMax* in O(log n) per operation. The structure is identical to a binary heap, but the heap-order property is that for any node, x, at even depth, the element stored at x is smaller than the parent but larger than the grandparent, and for any node x at odd depth, the element stored at x is larger than the parent but smaller than the grandparent.

a. How do we find the minimum and maximum elements?
b. Give an algorithm to insert a new node into the min-max heap.
c. Give an algorithm to perform *deleteMin* and *deleteMax*.

d. Can you build a min-max heap in linear time?

Q5.

a. Suppose we are given a sequence S of n elements, each of which is colored red or blue. Assuming S is represented by an array, give a linear-time in-place algorithm for ordering S so that all the blue elements are listed before all the red elements. What is the running time of your method?

b. Extend your approach from part (a) to k colors. That is, we are now given a sequence S of n elements, each of which is colored in one of k possible colors. Assuming S is represented by an array, give an in-place algorithm for ordering S so that all the elements of the same color appear consecutively. The algorithm should run in $O(nk)$ worst-case time.

Q6.

A group of children want to play a game, called Unmonopoly, where in each turn the player with the most money must give half of his/her money to the player with the least amount of money. What data structure(s) should be used to play this game efficiently? Why?