American University of Armenia
CS121 Data Structures
PSS 11

Heap

1. Given a HeapPriorityQueue, return the sum of the values between specified integer keys using default comparator.

2. Give an implementation of HeapPriorityQueue's downheap() method that uses recursion.

3. Given an array of digits (values are from 0 to 9), find the minimum possible sum of two numbers formed from digits of the array using a HeapPriorityQueue. (Note: All of the digits must be used.)

Heap Sort

1. Given an array of integers and an integer K, sort the elements of the given array in the increasing order of their modulo with K.
   Example:
   Initial array: [10, 3, 2, 6, 12]
   K = 4
   Sorted array: [12, 2, 6, 10, 3]

2. Given an array of n distinct integers, sort all even numbers in decreasing and odd numbers in increasing order. The modified array should contain all reverse sorted even numbers followed by sorted odd numbers.
   Example:
   Initial array: [1, 6, 8, 9, 3, 4, 2]
   Sorted array: [8, 6, 4, 2, 1, 3, 9]

Map

1. Given a string find the first K non-repeating characters using UnsortedTableMap.
   Example:
   String: ABCDBAGHCHFAC
   K = 3
   Output: D G F

2. Implement a method findKey() in UnsortedTableMap that given a value returns the first key associated with the specified value.

3. Extend UnsortedTabelMap<K, Integer> and override put(K, V) method so that if the given key already exists in the map, replace its value with the average of the new value and the current value stored in the key.
   Example:
   put(A, 5) Map: {(A5)}
   put(A, 31) Map: {(A18)}