Homework #5

Grading policy: Only P1-P5 will be graded. The others are for your practice.

Rules regarding the implementations of sorting algorithms:

Rule 1: Don't need to do insertion sort optimization, just use the simple algorithm.

Rule 2: Always use the first element as the pivot for quick sort. You can assume the rules are the default option in homework, quiz and exam, unless specified.

P1: Sort the following sequence [254, 564, 425, 628, 614, 021, 213, 117] step-by-step by using 1) merge sort, 2) quick sort, and 3) radix sort (in decimal base).

P2 Stable sorting: Sometimes we want the sorting to be stable, when the *key* can be repetitive.

A sorting algorithm is said to be stable if two objects with equal keys appear in the same order in sorted output as they appear in the input data set.

For the following algorithms, which ones are stable?

1) insertion sort, 2) heap sort, 3) merge sort, 4) quick sort.

P3: Illustrate step by step how heapsort sorts [5,13,2,25,7,17,20,8,4]. Remember you can do everything in an array. First heapify the array, then extraction the mins one by one and put the number in-place in the array.

P4 Inversion counting: Given an array of size N. Find the inversion count of the array in $O(N \log(N))$. Inversion is defined in *Lecture 11*. **Hint:** use merge sort.

P5 H-Index: Try to solve the *h-index* problem in linear complexity. You can assume the $0 \leq citation[i] \geq 1000$.

The h-index is an author-level metric that measures both the productivity and citation impact of the publications, initially used for an individual scientist or scholar. The h-index is defined as the maximum value of h such that the given author/journal has published at least h papers that have each been cited at least h times.

As an algorithm design problem:

Given an array of integers citations where citations[i] is the number of citations a researcher received for their i-th paper, return the researcher's h-index.

Example 1:

Input: citations = [3,0,6,1,5]

Output: 3

Explanation: [3,0,6,1,5] means the researcher has 5 papers in total and each of them had received 3, 0, 6, 1, 5 citations respectively. Since the researcher has 3 papers with at least 3 citations each and the remaining two with no more than 3 citations each, their h-index is 3.

Example 2:

Input: citations = [1,3,1]

Output: 1

P6: Chocolate Distribution Problem: Given an array of N integers where each value represents the number of chocolates in a packet. Each packet can have a variable number of chocolates. There are m students, the task is to distribute chocolate packets such that:

1. Each student gets one packet.
2. The difference between the number of chocolates in the packet with maximum chocolates and the packet with minimum chocolates given to the students is minimum.

Example:

Input: arr[] = 7, 3, 2, 4, 9, 12, 56 , m = 3

Output: Minimum Difference is 2

Explanation: We have seven packets of chocolates and we need to pick three packets for 3 students If we pick 2, 3, and 4, we get the minimum difference between maximum and minimum packet sizes.

P7: Leetcode 451. Sort Characters By Frequency

P8: Leetcode 218. The Skyline Problem

P9: Leetcode 1366. Rank Teams by Votes

P10: Leetcode 56. Merge Interval

Note: beyond this problem itself, interval is actually an commonly used data structure in many applications. For example, interval tree can be useful for scheduling and geometry.

P11: Leetcode 2251. Number of Flowers in Full Bloom