EL9343 Homework 2

(Due October 9th, 2020)

No late assignments accepted

All problem/exercise numbers are for the third edition of CLRS text book

---

1. For the maximum subarray problem, if we use divide-conquer, but instead of dividing the array into two halves, we equally divide it into three segments, how should the algorithm be modified? What is the running time of the new algorithm?

2. Demonstrate the operation of HOARE-PARTITION on the array A = <13, 19, 9, 5, 12, 8, 7, 4, 11, 2, 6, 21>. Show the values of the array after each iteration of the while loop in the lines of 4 to 11 in the code of lecture notes.

3. For the following array:

   A = <13, 19, 9, 5, 12, 8, 7, 4, 11, 2, 6, 21>

   (a) Create a max heap using the algorithm BUILD-MAX-HEAP.

   (b) Remove the largest item from the max heap you created in 3(a), using the HEAP-EXTRACT-MAX function. Show the array after you have removed the largest item.

   (c) Using the algorithm HAX-HEAP-INSERT, insert 56 into the heap that resulted from question 3(b). Show the array after insertion.

4. Let T be a min heap storing n keys. Given 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. Ideally, your algorithm should run in $O(k)$ time, where k is the number of keys reported that is smaller than x.

5. Using Figure 2.4 on page 35 of CLRS as a model, illustrate the operation of insertion sort and merge sort on the array [20, 17, 43, 25, 4, 72, 15, 64].

6. Design an algorithm to merge k sorted arrays, and return it as a new array. The new array

should be made by splicing together the nodes of the k arrays. Additionally, the total number of elements of all arrays is kn. (Notice that the number of elements of each array is not necessary the same). Ideally, your algorithm should run in $O(kn \log k)$ time, lower algorithm will lose some points. Please give the procedure of your algorithm and analyze the running time. (Description is enough, you do not need to provide any pseudocode)

For example:

Input: A: <1, 5, 18>, B: <3, 6, 7, 11>, C: <2, 4, 9, 12, 13>

Output: <1, 2, 3, 4, 5, 6, 7, 9, 11, 12, 13, 18>