EL9343 Homework 4

No late assignments accepted

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

---

1. For the following array:

$$A = <15, 22, 25, 17, 12, 19, 23, 16, 24, 14, 10, 26>$$

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

   (b) Design an algorithm to create a min heap. (Pseudocode is required)

   (c) Create a min heap using the algorithm you designed in 1(b)

   (d) Remove the largest item from the max heap you created in 1(a), using the HEAP-EXTRACT-MAX function. Show the array after you have removed the largest item.

   (e) Using the algorithm MAX-HEAP-INSERT, insert 11 into the heap that resulted from question 1(d). Show the array after insertion.

2. Design two different algorithms 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). One of your algorithms should run in $O(kn \log k)$ time. 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, 4, 7, 10>, B: <2, 5, 8, 11>, C: <3, 6, 9>

   Output: <1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11>

3. For the following array:

$$A = <1, 5, 9, 6, 3, 2, 8, 7, 4, 0>$$

   (a)   Illustrate the operation of quick sort on array A

   (b)   Illustrate the operation of merge sort on array A

   (c)   Explain the advantage and disadvantage of sorting an array by quick sort compared to using merge sort.

4. For an disordered array with n elements, design an algorithm for finding the median of this array. Your algorithm should traverse the array only once.

   ps:

You can imagine the array as a flow which means you can get the data one by one.

The size of this array --n is big and you know the size of the array from the start.

Please do not sort the array, or you can not get full mark.

A hint to solve this problem is using heap.