Homework 02

CS 624, 2024 Spring

1. Exercise 6.5-8 (page 166) on Heap-Delete.

   > The operation HeapDelete(A, i) deletes the item in node i from heap A. Give an implementation of HeapDelete that runs in $O(\lg n)$ time for an n-element max-heap.

   Include a brief explanation of the running time.

2. Exercise 6.5-9 (page 166) on merging k sorted lists.

   > Give an $O(n \lg k)$-time algorithm to merge k sorted lists into one sorted list, where n is the total number of elements in all the input lists. (Hint: Use a min-heap for k-way merging.)

   Include a brief explanation of the running time.

3. Exercise 6.1 in the Lecture 3 auxiliary handout on selecting k smallest elements.

4. Problem 7-2 (page 186) on quicksort with equal element values.

   Do parts (a) and (b) from the textbook. Do not do (c) or (d). Instead:

   (c′) Show the steps of your Partition′ algorithm on the input array $[1, 6, 5, 8, 5, 4, 5]$ with $p = 1$ and $r = 7$. That is, partition the entire array. (Similar to Figure 7.1 on page 172.)

   (d′) State the loop invariant for your Partition′ algorithm. You are *not* required to write the proof of correctness, but the loop invariant you state must be correct and it must be strong enough to prove the correctness of your algorithm.

   Your Partition′ procedure must not be randomized; it should use the final element of the array range as the pivot, like Partition does. It should make a single pass over the array range. I recommend using a **while** loop instead of a **for** loop, but it can be solved either way.

5. Problem 7-4 (page 188) on Tail-Recursive-Quicksort.

6. Problem 7-6 (page 188) on fuzzy sorting of intervals.