Problem Set 6

Data Structures and Algorithms, Fall 2019

Due: October 24, in class.

Announcement

It is almost the half point of the semester now, we (the lecturer and the TAs) hope you are enjoying this course, and would like to listen to your voice! So, if you have any comments or suggestions, feel free to share them with us. You can drop us a Email, or simply write down the comments/suggestions in your exercise book. We look forward to hearing from you! ☺

From CLRS

Exercise 12.1-5, 12.2-1, 12.3-3, 12.3-4. Problem 12-1. Exercise 13.1-4, 13.2-4, 13.3-2 (show the RB-tree after *each* insertion), 13.4-3 (show the RB-tree after *each* deletion). Problem 13-3.

Additional Problem One

[You do not need to solve both (a) and (b), choose one among the two.
But feel free to submit solutions for both.]

(a) Design an algorithm that constructs a Treap from a size n sorted array $A = (x_1, \cdots, x_n)$. This method should run in $O(n)$ worst-case time and should construct a Treap that is indistinguishable from one in which the elements of A were added one at a time using the INSERT(x_i) method. (That is, build a random Treap containing elements in A in $O(n)$ worst-case time.)

(b) Suppose we are given two skip lists, one storing a set A of m keys, and the other storing a set B of n keys. Describe and analyze an algorithm to merge these into a single skip list storing the set $A \cup B$ in $O(n + m)$ expected time. Do *not* assume that every key in A is smaller than every key in B; the two sets could be arbitrarily intermixed.