Department of Computer Science **University of San Francisco**

Computer Science 673
Fall 2016
Homework 5: Trees and Dynamic Programming
Due Friday, September 30th

All problem / exercise numbers are from the **3rd edition** of Introduction to Algorithms (the 1st and 2nd editions are different!) Note the difference between problems and exercises!

1. (6 points) Exercise 14.2-1 (MINIMUM, MAXIMUM, SUCCESSOR, PREDECESSOR) Show how, by adding pointers to the nodes, how to support each of the dynamic-set queries: MINIMUM, MAXIMUM, SUCCESSOR, PREDECESSOR in $O(1)$ wrose-case time on an augmented order-statistic tree. The asymptotic performance of other operations on order-statistic trees should not be affected.

2. (6 points) Exercise 14.3-6 (MIN-GAP) Show how to extend a red-black tree to support the operation MIN-GAP, which gives the magnitude of the difference of the two closest numbers in the tree. For example, if $Q = \{1, 5, 9, 15, 18, 22\}$ then MIN-GAP(Q) returns 18-15 = 3 since 15 and 18 are the two closest numbers in Q.

3. (4 points) Give pseudo-code for an algorithm that takes as input an interval tree T and an interval i, and returns *all* intervals in T that overlap i in time $O(\min(n, k \lg n)$ time, where k is the number of intervals in the output list and n is the number of intervals in the tree. Your solution should not modify the tree.

4. (8 points) A **palindrome** is a non-empty string over some alphabet that reads the same forward and backward. Example palendromes are all strings of length 1, civic, racecar, and aibohphobia (fear of palindromes)

   Give an efficient algorithm to find the longest palindrome that is a subsequence of a given input string. For example, given the input character, your algorithm should return carac. What is the running time of your algorithm?