

Homework 4

CS 430 Introduction to Algorithms
Spring Semester, 2014

1. **Problem 1** Give an example where quicksort requires $O(n^2)$ steps.
(10 pts)
2. **Problem 2.** Problem 4-6 (Pg 110) CLRS(3rd Edition) (20 pts)
3. **Problem 3.** Using the version of heapsort as defined in CLRS (Chapter 6-4), show an example where heapsort requires $\Omega(n \log n)$ steps. (10 pts)
4. **Problem 4.** Consider radix sort with numbers (using base 10) that are variable length. Show that you can output any number as soon as you have considered all its digits.
Design a method to sort in $O(n + k)$ time where k is the total number of digits in all the numbers.
(20 pts)
5. **Problem 5.** Suppose that you are given a sequence of n elements to sort. The input sequence consists of $\frac{n}{\log n}$ subsequences, each containing $(\log n)$ elements. The elements in a given subsequence are all smaller than the elements in the succeeding subsequence and larger than the elements in the preceding subsequence. Thus, all that is needed to sort the whole sequence of length n is to sort the $(\log n)$ elements in each of the $\frac{n}{\log n}$ subsequences. Show an $\Omega(n \log \log n)$ lower bound on the number of comparisons needed to solve this variant of the sorting problem. (Hint: It is not rigorous to simply combine the lower bounds for the individual subsequences.)
(20 pts)