Analysis of Algorithms: Assignment 4

Due date: February 4 (Thursday)

Problem 1 (3 points)
Consider a computer environment where the control flow of a program can split three ways after a single comparison $a_i : a_j$, according to whether $a_i < a_j$, $a_i = a_j$, or $a_i > a_j$. Argue that the number of these three-way comparisons required to sort an n-element array is $\Omega(n \lg n)$.

Problem 2 (3 points)
Using Figure 9.2 (page 176) in the textbook as a model, illustrate the operation of COUNTING-SORT on the array $A = \langle 7, 1, 3, 1, 2, 4, 5, 7, 2, 4, 3 \rangle$.

Problem 3 (4 points)
Briefly describe how to adapt (a) MERGE-SORT and (b) QUICK-SORT to sort elements stored in a linked list. Give the time complexity of your algorithms; is it the same as the complexity of sorting an array?

Problem 4 (bonus)
This problem is optional; if you solve it, then you get 2 bonus points toward your final grade for the course. You cannot submit this bonus problem after the deadline.

Consider an array $A[1..n]$ whose elements are distinct integer numbers, and describe an algorithm that finds the largest and second largest elements of this array using $(n + \lceil \log n \rceil - 2)$ comparisons.