Homework 5

You have to submit your solutions as announced in the lecture.
Unless mentioned otherwise, all problems are due 2017-03-16, 11:00.
There will be no deadline extensions unless mentioned otherwise in the lecture.

Problem 5.1 *Standard Libraries*

Points: 0

In your favorite programming language's standard library, track down the source code of

- (mutable or immutable) linked lists
- the sorting algorithm(s) for arrays or lists (Which one is it?)

Problem 5.2 *Sorting Lists of Integers*

Points: 8+8

We want to sort lists of integers according to $\leq$.

Implement 2 out of bubble sort, insertion sort, merge sort, and quick sort in any programming language(s).

Remark: Alternatively, you may implement the algorithms for arbitrary arguments $ord : TotOrd[A], x : List[A]$. (That's actually the same amount of work once you have understood polymorphism.)

Problem 5.3 *Sorting Lists of Strings*

Points: 8

Adapt one of your two sorting implementations from the previous question to sort lists of strings lexicographically.

Remark: If you gave a polymorphic solution above, these points are for free using $Lexicographic : TotOrd[string]$ from before. Otherwise, you have to duplicate your work.

Problem 5.4 *Comparing Sorting Algorithms*

Points: 5

Using the lecture notes, literature, the internet etc., make a table that compares as many sorting algorithms as you can find. For each algorithm, collect at least

- Θ-class of the best-case, average-case, and worst-case time complexity (measured in the length of the list)
- whether it can be used as an in-place algorithm

Remark: Such questions often occur in job interviews or in standardized tests, e.g., for grad school admission. So it is good to have such a table handy.