CS120: Intro. to Algorithms and their Limitations Anshu & Hesterberg

Sender–Receiver Exercise 0: Reading for Receivers

Harvard SEAS - Fall 2023 *2023-09-12*

The goals of this exercise are:

- to develop your skills at understanding, distilling, and communicating proofs and the conceptual ideas in them
- to practice reasoning about the running time of algorithms
- to see how the choice of a model of computation can affect the computational complexity of a problem

In the previous class, we have seen that there exists a sorting algorithm whose worst-case running time is $O(n \log n)$. In fact, the (worst-case) computational complexity of sorting by comparison-based algorithms is $\Theta(n \log n)$. That is, *every* (correct) comparison-based sorting algorithm (one that operates just by comparing keys to each other) has worst-case running time $\Omega(n \log n)$. This holds even when the keys are drawn from the universe $U = [n]$.

In our first active learning exercise, you will see that for keys drawn from the universe $[n]$, it is actually possible to sort asymptotically faster — in time $O(n)$! How is this possible in light of the $\Omega(n \log n)$ lower bound? Well, the algorithm will not be a comparison-based one; it will directly access and manipulate the keys themselves (rather than just comparing them to each other).

More generally, we will show:

Theorem 0.1. *There is an algorithm for sorting an array of n key-value pairs where the keys are drawn from a known universe of size U with (worst-case) running time $O(n + U)$.*

Since we have not yet precisely defined our computational model or what constitutes a "basic operation," this theorem and its proof are still informal. As you will see in Problem Set 2, it is possible to improve the dependence on U from linear to logarithmic with a more involved algorithm.

Proof. 1. Algorithm:

2. Correctness:

3. Runtime:

$\square$