6.897 ADVANCED DATA STRUCTURES (SPRING ’05)
Prof. Erik Demaine TA: Mihai Pătraşcu

Problem 7 Due: Wednesday, Mar. 30

Timing: This problem is due after spring break. In the spirit of not making you work during the break, we are making the problem due on a Wednesday, so you can decide to only look at it after school resumes.

Prove that on a word RAM with \( w \)-bit words, one can sort \( n \) \( w \)-bit integers in time \( O(n \lg \frac{w}{\lg n}) \). The algorithm can be randomized (the time bound must hold in expectation).

Hints: Think of van Emde Boas, and find a way to reduce sorting \( n \) integers of \( w \) bits to the problem of sorting \( n \) integers on \( \frac{w}{2} \) bits. Bottom out the recursion in a linear-time sorting algorithm (for \( w = \lg n \)). Note that you must reduce to a problem on exactly (or at most) \( n \) integers, not on \( O(n) \) integers (if you don’t see why, brush up on your recursions). The only randomness needed in the algorithm is through black-box use of hash tables.

In class, we saw that for \( w = \Omega(\lg^{2+\varepsilon} n) \), we can sort in linear time. In general, the sorting time drops quickly when \( w \) exceeds \( \lg^2 n \). This problem shows that the sorting time also drops quickly when \( w \) approaches \( \lg n \). Thus, the hardness of sorting is concentrated in a very narrow interval for \( w \): between \( \lg^{1+\varepsilon} n \) and \( \lg^2 n \). What happens in this interval remains an important open problem.