Assignment 5 (Bonus)

Grades: This assignment is 2% (bonus) in the final score. We use the following scale for the simplicity of grading: the total score is 20 for this assignment. Note the special submission time! No Late Submission is Accepted! Please submit on time for this bonus assignment.

Problem 1 [Full Score: 5]. Suppose we are given a sequence $S$ of $n$ elements, each of which is an integer in the range $[0, n^2 - 1]$. Describe a simple method for sorting $S$ in $O(n)$ time. HINT: Think of alternate ways of viewing the elements.

Problem 2 [Full Score: 15]. We say that an array $A[1 \cdots n]$ is $k$-sorted if it can be divided into $k$ blocks, each of size $n/k$, such that the elements in each block are larger than the elements in earlier blocks, and smaller than elements in later blocks. The elements within each block need not be sorted. For example, the following array is 4-sorted:

$[1, 2, 4, 3|7, 6, 8, 5|10, 11, 9, 12|15, 13, 16, 14]$

(1) Describe an algorithm that $k$-sorts an arbitrary array in $O(n \log k)$ time. (You should provide an algorithm and analyze its complexity. HINT: think about the quick-sort discussed in the lecture.)

(2) Use the decision tree method to prove that any comparison-based $k$-sorting algorithm requires $\Omega(n \log k)$ comparisons in the worst case. (HINT: count the number of possible outcomes of any $k$-sorting algorithm. Use the Stirling’s approximation $\log(n!) \sim n \log n - n$. )