Warm-up

Problem 1. Come up with an instance showing that SELECTION-SORT takes $\Omega(n^2)$ time in the worst case.

Problem 2. Come up with an instance showing that INSERTION-SORT takes $\Omega(n^2)$ time in the worst case.

Problem solving

Problem 3. Come up with an instance showing that HEAP-SORT takes $\Omega(n \log n)$ time in the worst case.

Problem 4. Given an array A with n integers, an inversion is a pair of indices $i < j$ such that $A[i] > A[j]$. Show that the in-place version of INSERTION-SORT runs in $O(n + I)$ time where I is the total number of inversions.

Problem 5. Given an array A with n distinct integers, design an $O(n \log k)$ time algorithm for finding the kth value in sorted order.

Problem 6. Given k sorted lists of length m, design an algorithm that merges the list into a single sorted lists in $O(mk \log k)$ time.