4. /6 Sorting Algorithms

Let $A[0 \ldots n-1]$ be filled with integers in the range $[0, n^2 - 1]$. Give your answers to the following questions using Θ notation in terms of n.

(a) What is the run-time of Merge-Sort on A?

(b) What is the run-time of Heap-Sort on A?

(c) Describe how to sort A in $O(n)$ time.

(d) What is the auxiliary space requirements of the algorithm from part (a)?

(e) What is the auxiliary space requirements of the algorithm from part (b)?

(f) What is the auxiliary space requirement of your algorithm from part (c)?

6. /4 Lower Bounds

Three-light-coin problem You have seven coins that look identical, but three of the coins are slightly lighter than the others. The three light coins all weigh the same, and the four heavy coins all weigh the same. You also have a balance scale that will compare the weights of any two subsets of coins and tell you either that the first subset weighs more, the second subset weighs more, or both subsets weigh the same. The problem is to determine which coins are light and which coins are heavy.

Prove the following theorem.

Theorem 1 *Any algorithm to solve the three-light-coin problem must use at least four weighings in the worst case.*

9. /4 Randomized Algorithms: Part 2

Your are given an array A of length n together with the dominant element x of A. Describe using pseudo-code a randomized algorithm *foo3*(A, x) that finds an index i such that $A[i] = x$. Your algorithm should have worst-case run-time $\Theta(n)$ and expected run-time $\Theta(1)$. The input array A should not be modified.

Justify the worst-case run-time and expected run-time bounds for your algorithm.