

HW Assignment 2 (Due date: September 30, Friday)

1. [Divide and Conquer, 20 points] Provide a divide-and-conquer algorithm for determining the **smallest and second smallest values** in a given unordered set of numbers. Provide a recurrence equation expressing the time complexity of the algorithm, and derive its **exact** solution (i.e., not the asymptotic solution). For simplicity, you may assume the size of the problem to be an exact power of a natural number.
2. [Divide and Conquer, 20 points] Describe a divide-and-conquer algorithm for computing a^n (where $n \in \mathcal{N}$) that runs in $\Theta(\lg n)$ time. Justify your answer.
3. [Heap, 20 points] Exercise 6.4-4, page 160. For simplicity, you may assume the size of the problem to be an exact power of a natural number.
4. [Heap, 10 points] Give pseudocode for `HEAP-DECREASE-KEY(A, i, key)` that runs in $O(\lg n)$ time for an n -element max-heap.
5. [QuickSort, 10 points] Exercise 7.2-1 page 178.
6. [QuickSort, 10 points] Write the pseudocode for the `PARTITION` algorithm studied in class.
7. (*) [Finding the missing integer, 20 points] An array $A[1..n]$ contains all the integers from 0 to n except one. It would be easy to determine the missing integer in $O(n)$ time by using an auxiliary array $B[0..n]$ to record which numbers appear in A . In this problem, however, we cannot access an entire integer in A with a single operation. The elements of A are represented in binary, and the only operation we can use to access them is “fetch the j th bit of $A[i]$ ”, which takes constant time. Show that if we use only this operation, we can still determine the missing integer in $O(n)$ time.