Reading Material

KT Chapter 4 (Divide and Conquer), section 4.1 (A First Recurrence: The Mergesort Algorithm), 4.2 (Further Recurrence Relations), and 4.3 (Counting Inversions).

If you have another edition of the book than the international one from 2014 Divide and Conquer might be in a different chapter (e.g. 5) , but the subsections are the same. In that case you should of course also solve the corresponding exercises (the numbers within the chapters are the same, e.g. if Divide and Conquer is chapter 5 in our book, you should solve exercise 5.1 instead of 4.1).

Exercises

Exercises marked with [w] are easy warmup exercises. They are just there to check that you understand the very basics. Exercises marked with [*] are extra difficult exercises. The puzzle of the week is just for fun.

1 Reccurrences I Use both the *recursion tree method* and the *(partial) substitution method* to solve each of the following recurrences.

1.1 $T(n) \leq \begin{cases} 2T(n/4) + cn & \text{for } n > 4 \\ c & \text{otherwise} \end{cases}$

1.2 $T(n) \leq \begin{cases} 2T(n/4) + c\sqrt{n} & \text{for } n > 4 \\ c & \text{otherwise} \end{cases}$

2 Significant Inversions Solve KT exercise 4.2.

3 Divide-and-conquer on trees Solve KT exercise 4.6.

4 Divide-and-conquer on grid graphs Solve KT exercise 4.7.

5 CSES exercises To get familiar with the CSES online judge system solve the following two exercises on CSES (you first need to make a CSES profile).

- Missing Number: https://cses.fi/problemset/task/1083
- Distinct Numbers: https://cses.fi/problemset/task/1621

We will use CSES for our programming exercises during the course.

6 Reccurrences II Use both the *recursion tree method* and the *(partial) substitution method* to solve each of the following recurrences.

6.1 $T(n) \leq \begin{cases} T(\frac{3n}{4}) + cn & \text{for } n > 4 \\ c & \text{otherwise} \end{cases}$

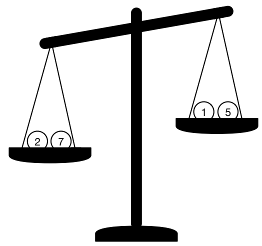
6.2 $T(n) \leq \begin{cases} T(n/2) + T(n/3) + T(n/6) + cn & \text{for } n > 6 \\ c & \text{otherwise} \end{cases}$

7 [$*$] **Divide-and-conquer on trees II** Consider the problem from exercise 2 (Divide-and-conquer on trees). What happens if the tree is not balanced? How many probes do we need in this case?

Hint: In a binary tree with n vertices there always exists a vertex such that if you remove it each of the remaining trees have size at most $n/2$. (This vertex is called a *separator*).

Finding such a separator does not count in the number of probes used in this exercise.

Puzzle of the week You are given twelve balls, eleven of which are identical and one of which is different, but it is not known whether it is heavier or lighter than the others. You have a traditional balance scale with two pans. To use such a scale, you can place a group of balls into each pan and the scale will determine which group is heavier.



The balance scale may be used three times to isolate the unique ball and determine wether it is heavier or lighter than the others. You can assume that the balls are numbered so that you can distinguish them from each other.