CIS 575. Introduction to Algorithm Analysis
Material for February 23, 2024

The Divide & Conquer Paradigm

The topic of this note is presented in *Cormen*'s Section 2.3.

1 Divide & Conquer

A very old strategy, in Latin called *divide et impera*, is: divide your enemies so as to easier conquer them!

General Principle In computer science, one can often solve a problem by the recipe:

1. decompose it into smaller instances;
2. solve (recursively) these instances;
3. combine these solutions into a solution to the original problem.

At some point, one cannot continue decomposing, but must solve the problem directly.

Analysis Assume that in the recipe above, we solve a problem directly when its "size" is below a certain threshold N, and otherwise split a problem into a smaller instances, each having a size that is b times smaller. With $f(n)$ the time it takes to decompose a problem of size n and to subsequently find its solution from the solution to the smaller instances, and $g(n)$ the time it takes to directly solve a problem of size n, the running time $T(n)$ of a divide & conquer algorithm will be given by the recurrences

$$\begin{array}{lcll} T(n) & = & aT(\frac{n}{b}) + f(n) & \text{for } n \geq N \\ T(n) & = & g(n) & \text{for } n < N \end{array}$$

where we know from our study of recurrences that

> the asymptotic behavior of T does **not** depend on g or N.

Accordingly, our focus will be on how to **decompose problems** and how to **combine solutions**. For any application, experimental studies may be used to find a suitable value of N, and a suitable approach for handling input smaller than N.