CS-E3190 Principles of Algorithmic Techniques

02. Recursive Algorithms – Graded Exercise

Please read the following **rules** very carefully.

- Do not consciously search for the solution on the internet.
- You are allowed to discuss the problems with your classmates but you should **write the solutions yourself**.
- Be aware that **if plagiarism is suspected**, you could be asked to have an interview with teaching staff.
- Each week the second exercise is an **individual exercise**, and the teaching staff will not give hints or help with them. You are allowed to ask for hints for the first exercise.
- In order to ease grading, we want the solution of each problem and subproblem to start on a **new page**. If this requirement is not met, **points will be deduced**.

1. **Fibonacci sequence.** The Fibonacci sequence is defined as

$$\begin{cases} F(1) = 1 \\ F(2) = 1 \\ F(n) = F(n-2) + F(n-1), \text{ when } n > 2 \end{cases}$$

   (a) (1p.) Prove by induction that for all $n \geq 2$

$$\begin{pmatrix} F(n) \\ F(n-1) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-2} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

   Hint: the zeroth power of a matrix is the identity matrix.

   (b) (2p.) Use part (a) to design a recursive algorithm using the divide and conquer approach to compute $F(n)$, $\forall n \in \mathbb{N}$. The time complexity of the algorithm should be $o(n)$, ie. faster than linear. You may assume that integer addition and multiplication take $O(1)$ time.

   (c) (2p.) Analyse the time complexity of your algorithm.

2. **Individual exercise: Binomial coefficients computation.** Recall that the binomial coefficient $\binom{n}{k}$ is the number of ways to choose an unordered subset of size k from a fixed set of size n. It can be computed with the following formula:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

   (a) (2p.) Binomial coefficients satisfy the following property when $n > k$ and $k \geq 1$.

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}.$$

   Use this formula to design a recursive algorithm for binomial coefficients.

   (b) (1p.) Give the recurrence relation that corresponds to the time complexity of this algorithm. *Hint: You are allowed to use multiple variables in the recurrence relation.*

(c) (2p.) Use the recurrence relation to compute the time complexity of the algorithm with respect to n. Prove that the complexity is in $o(n!)$.

Hint: For sufficiently large n *we have the approximation*

$$\binom{2n}{n} \sim \frac{2^{2n}}{\sqrt{n\pi}}.$$