

PS Algorithms and Data Structures 2026

Task sheet 11

Exercise 31

Given an undirected graph $G = (V, E)$, of which it is known that each node has at most d neighbors. We are looking for a coloring of the nodes so that each node has a different color than each of its neighbors. Describe a greedy algorithm that solves this problem using at most $d + 1$ colors.

The chromatic number of the graph $\chi(G)$ gives the minimum number of colors necessary to color the graph. Prove or disprove that the algorithm you have described always produces an optimal result. So either show that the number of colors you need is always equal to $\chi(G)$, or find a counterexample where your algorithm requires more than $\chi(G)$ colors.

Exercise 32

Calculate an optimal prefix coding using Huffman for the following text, i.e., provide the code tree and the codes for the individual symbols.

braunschweigerbierbrauerburschenbrauenbrausigesbraunschweigerbraunbier

Exercise 33

Dynamic programming can be used to avoid costly recursions. Examples for this are:

- The computation of Fibonacci numbers:

$$F_i = \begin{cases} i & \text{if } i \leq 1 \\ F(i-1) + F(i-2) & \text{if } i > 1 \end{cases}$$

- The computation of the **length** of the longest common subsequence (LCS, see lecture).

How can these recursions be implemented memory-efficiently using dynamic programming? (**Example:** Computing the length of the LCS of two strings of length n can be implemented with $n + O(1)$ additional memory.)