PS Algorithms and Data Structures 2026

Task sheet 12

Task 34

The algorithm for LCS presented in the lecture computes not only the length of the subsequence but also the **subsequence itself**, but requires $\Theta(n^2)$ additional memory. How can you use your algorithm from exercise 33 from task sheet 11 to compute the subsequence with $n + O(1)$ additional memory?

Hint: The algorithm will have a worse time complexity.

Note: In general, different tradeoffs are possible here. For example, it is possible to compute the subsequence in $O(n^{\frac{5}{2}})$ time and $O(n^{\frac{3}{2}})$ space.

Task 35

The algorithm discussed in the VO for calculating the longest common subsequence can also be used without the additional information (the arrows) about the predecessor positions. Specify the algorithm in pseudocode that determines the longest common subsequence using the two sequences X, Y and the complete matrix C.

Task 36

A logistics company wants to optimize the use of its truck fleet for freight transport between Salzburg and Hamburg Altona. Their customers expect m Euro pallets (standardized size) to reach the loading point. For this purpose, the logistics company has a fleet $L_1, L_2, \ldots, L_n$ of n trucks of different types, which can be deployed as required. The jth truck has a loading capacity of k_j Euro pallets and, if used, generates flat-rate operating costs of b_j euros. For scheduling reasons, each truck in the fleet can only be used once.

We are looking for an algorithm that uses the dynamic programming method to determine the minimum total operating costs required to transport all m Euro pallets to the loading point with the available fleet. You can assume that the total capacity of the fleet is sufficient to transport all m Euro pallets.

1. Denote $S[i, j]$ the minimum total cost of ownership required to transport i pallets to the loading point using only the trucks $L_1, L_2, \ldots, L_j$. Specify a recursive formulation for $S[i, j]$ and prove the correctness of your recursive definition. Make sure to define all cases of the recursion base completely and unambiguously!
2. Let m and two arrays $K = [k_1, k_2, \ldots, k_n]$ and $B = [b_1, b_2, \ldots, b_n]$ of length n be given, where k_j is the capacity and b_j is the operating cost of the jth truck. Write an algorithm in pseudocode which, given (m, K, B, n), determines the minimum total costs using the

dynamic programming method. The specific selection of trucks does not have to be output explicitly.

3. Analyze the runtime of your algorithm.
4. How does your algorithm change if an additional car has to accompany the convoy with 2 or more trucks and this car causes fixed costs of C?