Algorithms - Exercise 4

Due Wednesday 17/11 24:00

1. Write down explicitly (including the tables) the run of the matrix-chain multiplication algorithm discussed in class on the following input:

matrix	dimension
A_1	25×40
A_2	40×10
A_3	10×30
A_4	30×15

2. Give a dynamic-programming solution to the activity selection problem (lots of activities, one lecture hall).

3. Give a dynamic-programming solution to the 0-1 knapsack problem that runs in time $O(nW)$, where n is the number of items and W is the maximal weight the thief is allowed to carry.

4. (a) Give an $O(n^2)$ algorithm to find the longest monotonically increasing subsequence of a sequence of n numbers.

   (b) (Reshut) Give an $O(n * \log n)$ algorithm to find the longest monotonically increasing subsequence of a sequence of n numbers.

5. Let T be a tree and w a weight function on the nodes of T. Show how to find a subset of the nodes with maximal weight, such that no two adjacent nodes are chosen to be in the subset.

6. (a) Give a formal definition of fully parenthesized product of matrices.

   (b) Write the recursive formula for the number of possibilities to fully parenthesize a product of n matrices. Prove that the solution to this formula is $\Omega(2^n)$

   (c) (reshut) Prove that the solution to the formula above is $\Omega(4^n/n^{3/2})$.