ADS practice questions: Dynamic programming

Question 1.

A tree trunk of length L is to be cut into logs of different lengths in a manner that minimizes waste. The lengths of $n \geq 0$ logs are $k_1, \ldots, k_n$, where $k_i \geq 0$ for all $0 < i \leq n$ (assume that both L and k_i are integers, for example centimeters). Find a subcollection of $\{ k_1, \ldots, k_n \}$ such that the logs can be cut from the trunk, and with the minimum possible amount of waste. Within this context 'waste' is understood as a remaining section of trunk from which it is no longer possible to cut a log in the specified subcollection that has not already been cut. You may assume, for convenience, that the blades of the chainsaw used to cut the logs have a width of 0 cm.

An alternative and slightly more precise specification of the problem is: minimize $L - \sum_{i=1}^{n} b_i \cdot k_i$ subject to the condition that $\sum_{i=1}^{n} b_i \cdot k_i \leq L$ en $b_i \in \{ 0, 1 \}$ for all $0 < i \leq n$.

Questions:

1. Determine which of the following 8 lengths of logs can be cut from a tree trunk of a length 100 cm with a minimum amount of waste. Explain why your solution minimizes waste. The lengths of the logs are: $k_1 = 10, k_2 = 12, k_3 = 20, k_4 = 30, k_5 = 53, k_6 = 10, k_7 = 5$ and $k_8 = 42$ (all dimensions are in centimetres).
2. Give a recurrence relation $C(i, l)$ for $i \geq 0$ that that describes the length of the minimum amount of waste left on cutting logs from the collection $\{ 1, \ldots, i \}$ from a tree trunk of length l. *Hint: It will be useful to assume that $C(i, l) = \infty$ for $l < 0$.*
3. Give an algorithm that determines the length of the minimum amount of waste from a tree trunk of length $L \geq 0$ and $n \geq 0$ logs of lengths $k_1, \ldots, k_n$. *Hint: use dynamic programming.*
4. Determine the worst case time complexity and spatial complexity of your algorithm.

Question 2.

Suppose that there are four matrices A, B, C and D with dimensions of 20x2, 2x15, 15x40 and 40x4. Give the *cost* matrix as computed by the algorithm in slide 27.

Question 3.

Suppose that you have the following keys, together with the probability that a search will be carried out for the relevant key: A .20, B .24, C .16, D .28, E .04, F .08. Give the *cost* matrix as computed by the algorithm in slide 33. What is the optimum binary search tree?

Question 4.

Suppose that the currency of a given country is made up of the series $c_1 > c_2 > \ldots > c_n$ (for example 50, 25, 10, 5, 1 for the USA). The *coin change problem* is: given a currency and change of a cents, what is the minimum number of coins that add up to a (assume that $c_n = 1$)?

1. Formulate a greedy algorithm for this problem. Show how this works for \$ 1.43.
2. State a currency of a fictitious country for which the greedy algorithm does not always yield the minimum.
3. State an algorithm (with met dynamic programming) that solves the problem with every currency.