DP - Mock Exam

Theory Task T3. / 9 P

In this problem, you are given an array $A = [a_1, \ldots, a_n]$ of n pairwise distinct positive integers, i.e. $a_i \in \mathbb{Z}_{\geq 1}$ for $i \in \{1, \ldots, n\}$ and $a_i \neq a_j$ for $i \neq j \in \{1, \ldots, n\}$.

/ 7 P

a) Provide a *dynamic programming* algorithm that, given an array A of n pairwise distinct positive integers as above, returns `True` if there are two different (non-empty) subsets I_1 and I_2 of $\{1, \ldots, n\}$ (i.e. $\emptyset \neq I_1, I_2 \subseteq \{1, \ldots, n\}$ and $I_1 \neq I_2$) such that $\sum_{i \in I_1} a_i = \sum_{i \in I_2} a_i$ and `False` otherwise. In particular, the algorithm does *not* have to return the sets I_1 and I_2.

For example,

- For the array $[2, 3, 4, 5, 7]$ the output should be `True` since for $I_1 = \{1, 2, 4\}$ and $I_2 = \{2, 5\}$ we have $\sum_{i \in I_1} a_i = 2 + 3 + 5 = 10 = 3 + 7 = \sum_{i \in I_2} a_i$.
- For the array $[2, 3, 4, 10, 20]$ the output should be `False` since there are no two different non-empty subsets I_1 and I_2 of $\{1, 2, 3, 4, 5\}$ that satisfy $\sum_{i \in I_1} a_i = \sum_{i \in I_2} a_i$.

In order to obtain full points, your algorithm should run in time $O(n \cdot S)$, where $S = \sum_{i=1}^{n} a_i$. In your solution, address the following aspects:

1. *Dimensions of the DP table*: What are the dimensions of the *DP* table?
2. *Subproblems*: What is the meaning of each entry?
3. *Recursion*: How can an entry of the table be computed from previous entries? Justify why your recurrence relation is correct. Specify the base cases of the recursion, i.e., the cases that do not depend on others.
4. *Calculation order*: In which order can entries be computed so that values needed for each entry have been determined in previous steps?
5. *Extracting the solution*: How can the solution be extracted once the table has been filled?
6. *Running time*: What is the running time of your solution?

Hint: It might be helpful to think about the following more general problem: For any integer $1 \leq m \leq S$, determine how many different subsets $I \subseteq \{1, \ldots, n\}$ there are such that $m = \sum_{i \in I} a_i$.

Dimensions of the DP table:

Scheme continues on the next page.

Subproblems:

Recursion:

Calculation order:

Scheme continues on the next page.

Extracting the solution:

Running time:

/ 2 P

b) Show that for any array A as above, the following two conditions are equivalent.

i) There are non-empty sets $I_1, I_2 \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in I_1} a_i = \sum_{i \in I_2} a_i$ and $I_1 \neq I_2$.

ii) There are non-empty sets $I_1, I_2 \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in I_1} a_i = \sum_{i \in I_2} a_i$ and $I_1 \cap I_2 = \emptyset$.