DGA Ü8

5) $n \in \mathbb{N}$ $\quad 1 = d_1 < d_2 < \ldots < d_k$ gesucht ist Summe aus $d_1, \ldots, d_k$, die n ergibt mit möglich wenigen Summanden

a) $b \leq n$ bel., sodass b als Teilsumme in der optimalen Summendarstellung von n vorkommt.

$$n = \sum_{j=1}^{m} d_{f(j)} = \sum_{j=1}^{l} d_{f(j)} + \sum_{j=l+1}^{m} d_{f(j)} \quad \text{mit } m \text{ minimal}$$

$f: \{1, \ldots, m\} \to \{1, \ldots, k\}$, sodass $n = \sum_{j=1}^{m} d_{f(j)}$ und $b = \sum_{j=1}^{l} d_{f(j)}$

Angenommen die Darstellung von b ist nicht optimal, d.h.

$\exists\, \tilde{l} < l \quad \exists\, g: \{1, \ldots, \tilde{l}\} \to \{1, \ldots, k\} : \sum_{j=1}^{\tilde{l}} d_{g(j)} = b$

$\Rightarrow \sum_{j=1}^{\tilde{l}} d_{g(j)} + \sum_{j=l+1}^{m} d_{f(j)} = n$ und die Summe ist um $l - \tilde{l} > 0$ kürzer $\lightning$

zu $n = \sum_{j=1}^{m} d_{f(j)}$ ist optimal

b) $m(n)$ ... Anzahl der Münzen im optimalen Fall

ges: Rekursion für $m(n)$

$\forall x < 0 : m(x) = +\infty$ $\quad ; \quad m(0) = 0 \quad ; \quad \forall x > 0 : m(x) = \min\{m(x - d_l) : l \in \{1, \ldots, k\}\} + 1$

c) ges: Algorithmus, der optimale Summe sucht

```
Algorithm(n) {
    if (n<0){return +∞;}
    if (n==0){return 0;}
    min = -∞;  min_l = -∞;
    for l = 1,...,k {
        tmp = Algorithm(n - d_l);
        if (tmp<min){min=tmp; min_l = l;}
    }
    print(min_l+1); return min;
}
```

Aufwand ist n^k, da es in jedem der maximal n Schritte k Münzen "zur Auswahl" gibt. Maximal n Schritte, da wenn man die Münze $d_1 = 1$ wählt n Rekursionsaufrufe benötigt.