Tutorial 3: Dynamic Programming

Problem 1. *Given an unlimited supply of coins of denominations $x_1, x_2, \ldots, x_n$ (where $x_1, \ldots, x_n$ are positive integer numbers), we wish to make change for a value v; that is, we wish to find a set of coins whose total value is v (the set may contain several coins of the same denomination). This might not be possible: for instance, if the denominations are 5 and 10 then we can make change for 15 but not for 12. Design a dynamic programming algorithm, with running time $O(nv)$, that does the following.*

1. *The algorithm determines if there is a set of coins of total value v.*
2. *If there is such set, the algorithm finds the set with the minimal possible number of coins.*

Describe your algorithm in detail. Prove its correctness.

Solution. Let $C(u)$ be the minimum number of coins the values of which sum up to u. $C(0) = 0$; in case u cannot be made using the given coins, we set $C(u) \coloneqq \infty$. We have that for $u > 0$,

$$C(u) = \min \{1 + C(u - x) : x \in \{x_1, \ldots, x_n\} \ \ \& \ \ x \leq u\},$$

where we take the minimum of an empty set to be ∞. To see this, note that if we insist the coin with value $x_i \leq u$ to be used to make u, then it must be that the minimum number of coins to make u is $1 + C(u - x_i)$. Then, raising that restriction, we get that the minimum number of coins needed to make u is $\min \{1 + C(u - x) : x \in \{x_1, \ldots, x_n\} \ \ \& \ \ x \leq u\}$.

A dynamic programming algorithm implementing this recursion scheme is:

$S[0] \coloneqq \emptyset$; $\mathrm{DP}[0] \coloneqq 0$;
for u **from** 1 **to** v
  if $u < \min\{x_1, \ldots, x_n\}$
    $\mathrm{DP}[u] \coloneqq \infty$;
  else
    $\mathrm{DP}[u] \coloneqq \min \{1 + \mathrm{DP}[u - x] : x \in \{x_1, \ldots, x_n\} \ \ \& \ \ x \leq u\}$;
    $x_i \coloneqq \arg\min \{1 + \mathrm{DP}[u - x] : x \in \{x_1, \ldots, x_n\} \ \ \& \ \ x \leq u\}$;
    $S[u] \coloneqq S[u - x_i] \cup \{x_i\}$;

We show that for each u, $DP[u]$ is indeed equal to $C(u)$ and $S[u]$ is a set of minimum size whose values add up to u, by induction on u.

Base case: If $u = 0$, then $\mathrm{DP}[u] = C(u)$ and $S[u] = \emptyset$. If $0 < u < \min\{x_1, \ldots, x_n\}$, then we cannot make u with the given coins and the algorithm sets $\mathrm{DP}[u] = \infty$ and leaves $S[u]$ unset.
Inductive step: If $u \geq \min\{x_1, \ldots, x_n\}$, then since

$$C(u) = \min\left\{1 + C(u - x) : x \in \{x_1, \ldots, x_n\} \ \ \& \ \ x \leq u\right\},$$

and by the induction hypothesis $\mathrm{DP}[u-x] = C(u-x)$, we have that $\mathrm{DP}[u] = C(u)$. Moreover, if x_i minimizes $1 + C(u - x)$ and $1 + C(u - x_i)$ is not ∞, then by the induction hypothesis $S[u - x_i]$ is a set of minimum size whose values add up to $u - x_i$, and $S[u] = S[u - x_i] \cup \{x_i\}$ is a set of minimum size whose values add up to u. $\square$