5/26/19 Author: Idan Damri

Minimal number of coins to pay for $X \in \mathbb{N}$ amount of money (we assume $0 \in \mathbb{N}$)

Coins are represented as a sequence $C_n = (c_1, c_2, \dots, c_n)$

Case 1:

Any c_i is a multiple of some c_j $(i \neq j)$
In this case, a greedy algorithm solves the problem.
We always take the next largest coin we allow to use

Case 2

Case 2 is more interesting, were their exists some c_i that don't have c_j $(i \neq j)$ s.t c_i is a multiple of c_j
we'll solve this by using dynamic programming

Assuming we're given X and C_n

If $X < 0$ then there's no $k \in \mathbb{N}$ that can be the solution to the give problem
so theoretically we can say that the answer is ∞

[redacted]

Else if $X = 0$ then the answer is 0

else $\forall i$ we subtract c_i from x and count 1 for using coin c_i and add to that the minimum result from using all other coins (one at a time) starting from the new amount $x - c_i$

Therefore we can say that

$$f(x) = \begin{cases} \infty, & x < 0 \\ 0, & x = 0 \\ 1 + \min\left(f(x - c_i)\ \forall_{1 \le i \le n}\right), & \text{otherwise} \end{cases}$$