TCSS 343 - Challenge - Greedy Coin Change

Jake McKenzie

August 4, 2018

Theorem. The set $S = \{1, 5, 10, 25\}$ is canonical. Let us define what we mean by canonical. A set is canonical if it satisfies the equation, where W is the least amount of change given back by a cashier:

$$W = \sum_{i=1}^{n} \left\lfloor \frac{D_i}{s_i} \right\rfloor$$

D_{n-1} is given by the following recursive definition:

$$D_{n-1} = D_n \% s_n$$

Where $D_n =$ "Whatever amount of change you start with" and % is denoted by the "remainder of".

My first intuition in proving this was exhaustion. I can write some code which will test for every case quite simply. First I had to find all optimal solutions must satisfy the following equations.

$$D_4 \leq \infty, D_3 + D_2 \leq 2, D_2 \leq 1, D_1 \leq 4$$

Because you can only ever give back at most 99 cents. The first can be found by using floored division to see at most what is each coin bounded by: $\left\lfloor \frac{24}{10} \right\rfloor = 2$ and the $\left\lfloor \frac{9}{5} \right\rfloor = 1$.

This "floored division" technique was first introduced by Don Knuth in the art of computer programming.

Let's use proof by infuction for the theorem above.
The **base case** is trivial. If you have zero coins the optimal solution is return zero.
For this reason we will make our **inductive hypothesis**: The greedy approach solves optimally for any value W such that $s_{k-1} \leq W < s_k$

Consider the optimal way of choosing change $s_{k-1} \leq W < s_k$ where the greedy approach always takes coin $k - 1$.

We claim that any optimal solution must also take $k - 1$ if not it needs enough coins of type $s_1, \ldots, s_{k-2}$ to add up to W. The floored division I showed implies that this is not possible. The problem reduces to solving for $W - s_{k-1} \to W$ such that $s_k \leq W < s_{k+1}$. Let us now apply our **induction step**: By introducing the inductive hypothesis and base case we find that $0 \leq W - s_{k-1} < s_k - s_{k-1}$

Let us now do a short proof by exhaustion with aid of a programming language. If I had more time I would use more rigorous proof software. I like Coq personally but for the interest of time I will use mathematica since this is the summer.

$$\begin{aligned} b &= \operatorname{mod}(a, 25) \\ c &= \operatorname{mod}(b, 10) \\ d &= \operatorname{mod}(c, 5) \end{aligned}$$

Then running the line Table[{Floor[$\frac{a}{25}$]$\leq$ ∞,Floor[$\frac{b}{10}$]+Floor[$\frac{c}{5}$] $\leq$ 2,d $\leq$ 4},{a,1,99}] I obtain the truth table below:

$$\begin{pmatrix} \text{True} & \text{True} & \text{True} \\ \text{True} & \text{True} & \text{True} \\ \text{True} & \text{True} & \text{True} \\ \text{True} & \text{True} & \text{True} \\ \text{True} & \text{True} & \text{True} \\ \text{True} & \text{True} & \text{True} \\ \text{True} & \text{True} & \text{True} \\ \text{True} & \text{True} & \text{True} \\ \text{True} & \text{True} & \text{True} \\ \vdots & \vdots & \vdots \\ \text{True} & \text{True} & \text{True} \\ \text{True} & \text{True} & \text{True} \\ \text{True} & \text{True} & \text{True} \\ \text{True} & \text{True} & \text{True} \\ \text{True} & \text{True} & \text{True} \\ \text{True} & \text{True} & \text{True} \\ \text{True} & \text{True} & \text{True} \\ \text{True} & \text{True} & \text{True} \\ \text{True} & \text{True} & \text{True} \\ \text{True} & \text{True} & \text{True} \end{pmatrix}$$