1. The **money changing problem** starts with a given set of positive integers called *denominations* $d_1, d_2, \ldots, d_n$ (think of them as the integers 1, 5, 10, and 25) and an integer A, we want to find nonnegative integers $a_1, \ldots, a_n \geq 0$ such that

$$A = \sum_{i=1}^{n} a_i d_i.$$

2. First, we note that A can be expressed as a linear combination of the d_i *if and only if* $d_i = 1$ for some i. Here is a proof.

   If one of your denominations d_i is 1, you will certainly be able to express every integer A as $\sum_{i=1}^{n} a_i d_i$ for some nonnegative integers $a_1, \cdots, a_n$. Conversely, in order to express $A = 1$ as a linear combination, you must have $d_i = 1$ for some i.

3. In general a necessary condition that $A = \sum_{i=1}^{n} a_i d_i$ is that $g = gcd(d_1, ..., d_n)$ divides A. In fact, $g|A$ turns out to be both necessary and sufficient for $A \geq X$ for some (large) X. Here is a proof.

   From the extended Euclidean algorithm we know we can write $g = \sum_{i=1}^{n} g_i d_i$ with some possibly negative g_i. Now let

$$\begin{aligned} G &= \sum_{i=1}^{n} |g_i| d_i, \\ d_{min} &= \min_i d_i, \\ k &= d_{min}/g, \\ X &= kG. \end{aligned}$$

   First note that the k consecutive multiples of g in the set $S = \{kG, kG + g, kG + 2g, \ldots, kG + (k-1)g\}$, all have nonnegative coefficients when written as $\sum_{i=1}^{n} a_i d_i$. The next multiple of g is $kG + kg = kG + d_{min}$, which has even larger nonnegative coefficients than kG. The next $k - 1$ multiples of g consequently also have nonnegative coefficients until we get to $kG + 2x_{min}$, and so on.

   Note that the coefficients are not necessarily unique (all the d_i could be identical), but we have shown that there is at least one set of nonnegative coefficients for all multiple of g at least equal to X.

4. The **optimal money changing problem** is that for a given A, find the nonnegative a_i's that satisfy $A = \sum_{i=1}^{n} a_i d_i$, and such that the sum of all a_i's is minimal — that is, you use the smallest possible number of coins.

5. Here is a *greedy algorithm* for solving this problem:

   Order your denominations such that $d_1 > d_2 > \cdots > d_n$. Then the *greedy algorithm* for this problem would be: Given A, let a_1 be the largest integer such that $a_1 d_1 \leq A$. If $A - a_1 d_1 > 0$, let a_2 be the largest integer such that $a_2 d_2 \leq A - a_1 d_1$. If you have nothing left over after doing this for $i = 1, \cdots, n$, then $A = \sum_{i=1}^{n} a_i d_i$.

6. Let us show that the greedy algorithm finds the optimum a_i's in the case of the denominations $\{1, 5, 10, 25\}$. Here is a proof.

   Since 1 divides 5 and 5 divides 10, it is clear that if we have a case in which the greedy algorithm would not find the optimal solution, it must involve 25, *i.e.* A must be greater than 25. Assume the greedy algorithm does not find the optimal solution for A, $A > 25$. Then $A = \sum_{i=1}^{4} a_i d_i = \sum_{i=1}^{4} b_i d_i$ and $\sum_{i=1}^{4} a_i > \sum_{i=1}^{4} b_i$, where the a_i were determined by the greedy algorithm and the b_i are optimal in that $\sum_{i=1}^{4} b_i$ is minimal. W.l.o.g. $a_4 = b_4$ [since $a_4 \leq 4$ any change of the number of 1 cent coins must occur in 5 unit steps to give the same sum-this is obviously worse than changing b_3], in addition to that note that $a_3 \leq 1$.

   By the above considerations we must have $a_1 > b_1$. Let $x := a_1 - b_1$. We have three cases to consider: $a_2 = b_2$, $a_2 > b_2$ and $a_2 < b_2$. If we set $y := a_2 - b_2$ then we can compute $b_3 = 5x + 2y + a_3$. Thus the number of coins changes by $\sum_{i=1}^{4} b_i - \sum_{i=1}^{4} a_i = 4x + y$. If we can show that this number is positive, this is a contradiction and we are done. In cases 1 and 2, x and y are ≥ 0. Therefore $4x + y$ is clearly positive.

   In case 3, y is negative. But, as we have to ensure that $b_3 = 5x + 2y + a_3$ is ≥ 0 and we know that a_3 is at most 1, we have $y \geq -\frac{5}{2}x - \frac{1}{2}$. Hence $4x + y \geq \frac{3}{2}x - \frac{1}{2}$ and it is again positive.

7. You can extend this problem and ask "*What are good necessary and sufficient conditions on a currency such that the greedy algorithm always gives the minimum amount of coins.*" This problem is still open. Partial answers and light hearted discussions can be found in the following references:

   (a) M. J. Magazine, G. L. Nemhauser, L. E. Trotter Jr., When the Greedy Solution Solves a Class of Knapsack Problems, Operations Research 23 (1975), p. 207 – 217

   (b) John Dewey Jones, Orderly Currencies, American Mathematical Monthly 101 (1994), p. 36 – 38

   (c) Stephen B. Maurer, Disorderly Currencies, American Mathematical Monthly 101 (1994), p. 419.