Coin Changing Problem

Greedy Algorithm to find Minimum number of Coins:

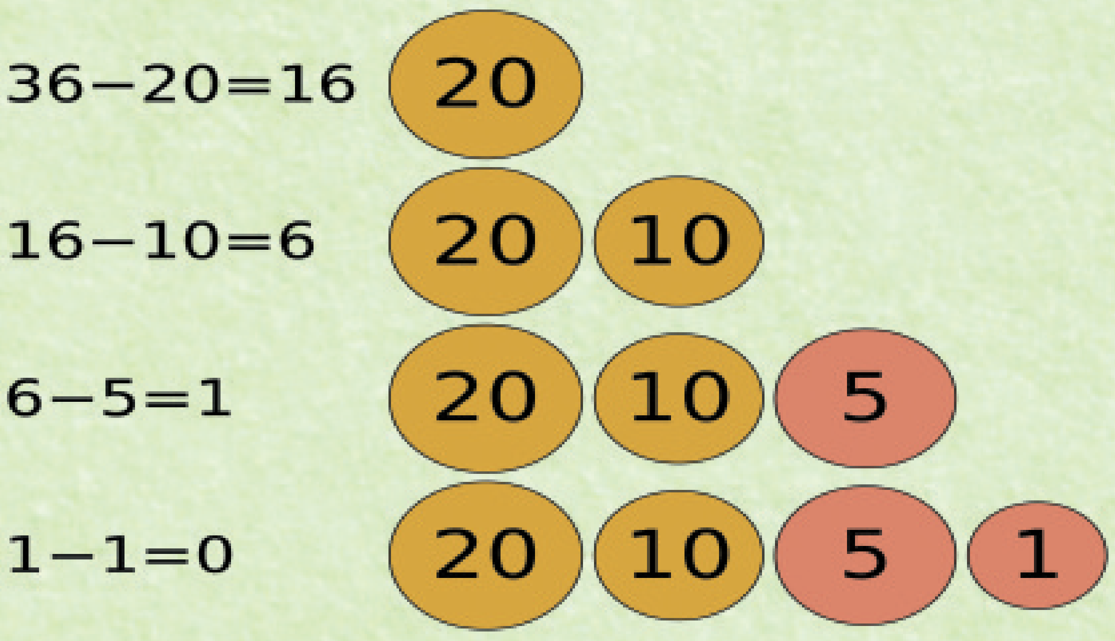
We have infinite supply of { 1, 2, 5, 10, 20, 50, 100, 500, 1000} valued coins/notes, what is the minimum number of coins and /or notes needed to make the change?

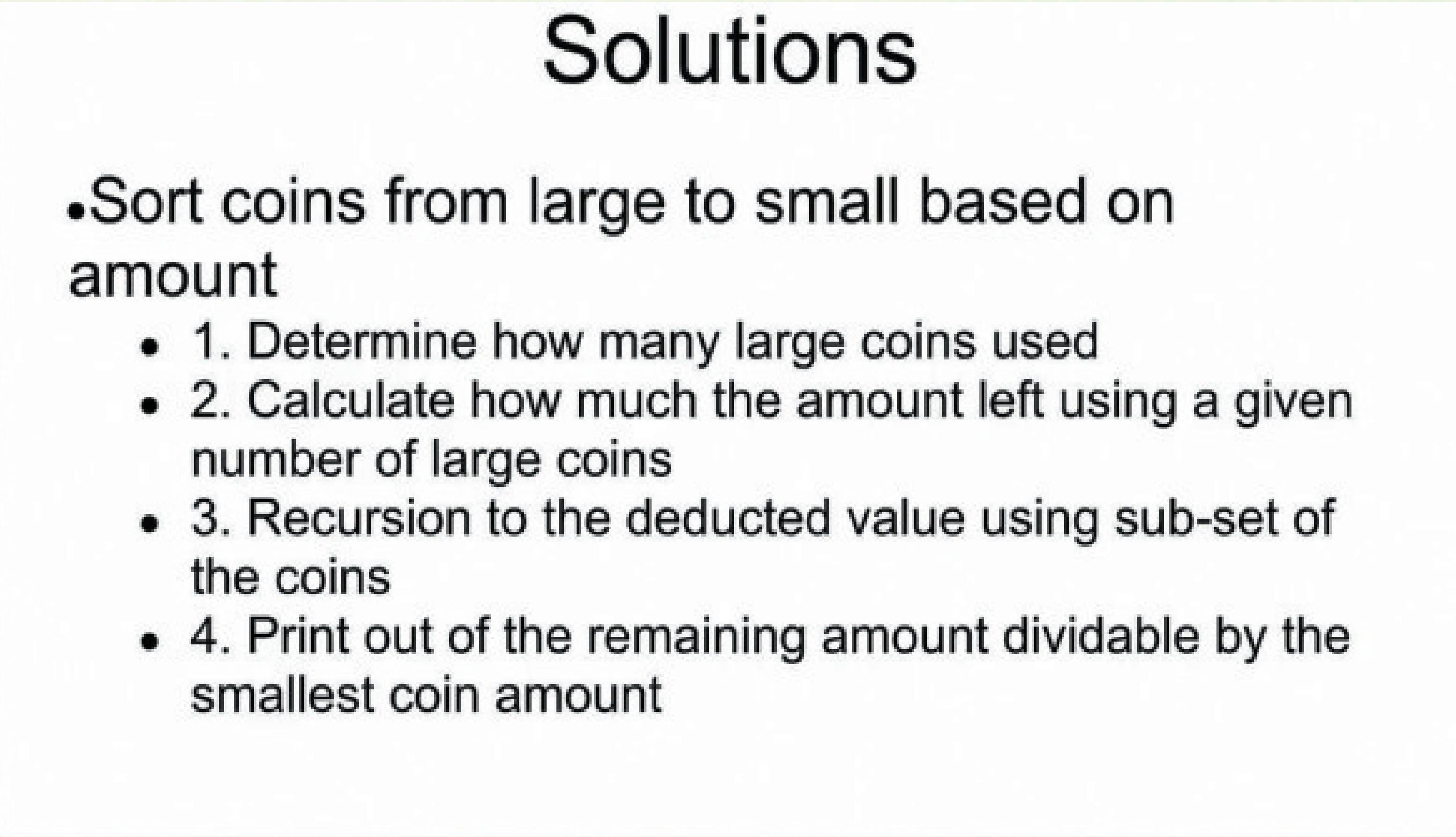
Input: V = 121
Output: 3
We need a 100 Rs note, a 20 Rs note and a 1 Rs coin.

Dynamic Programming | (Coin Change)

Given a value N, if we want to make change for N cents, and we have infinite supply of each of S = { S1, S2, .. , Sm} valued coins, how many ways can we make the change?

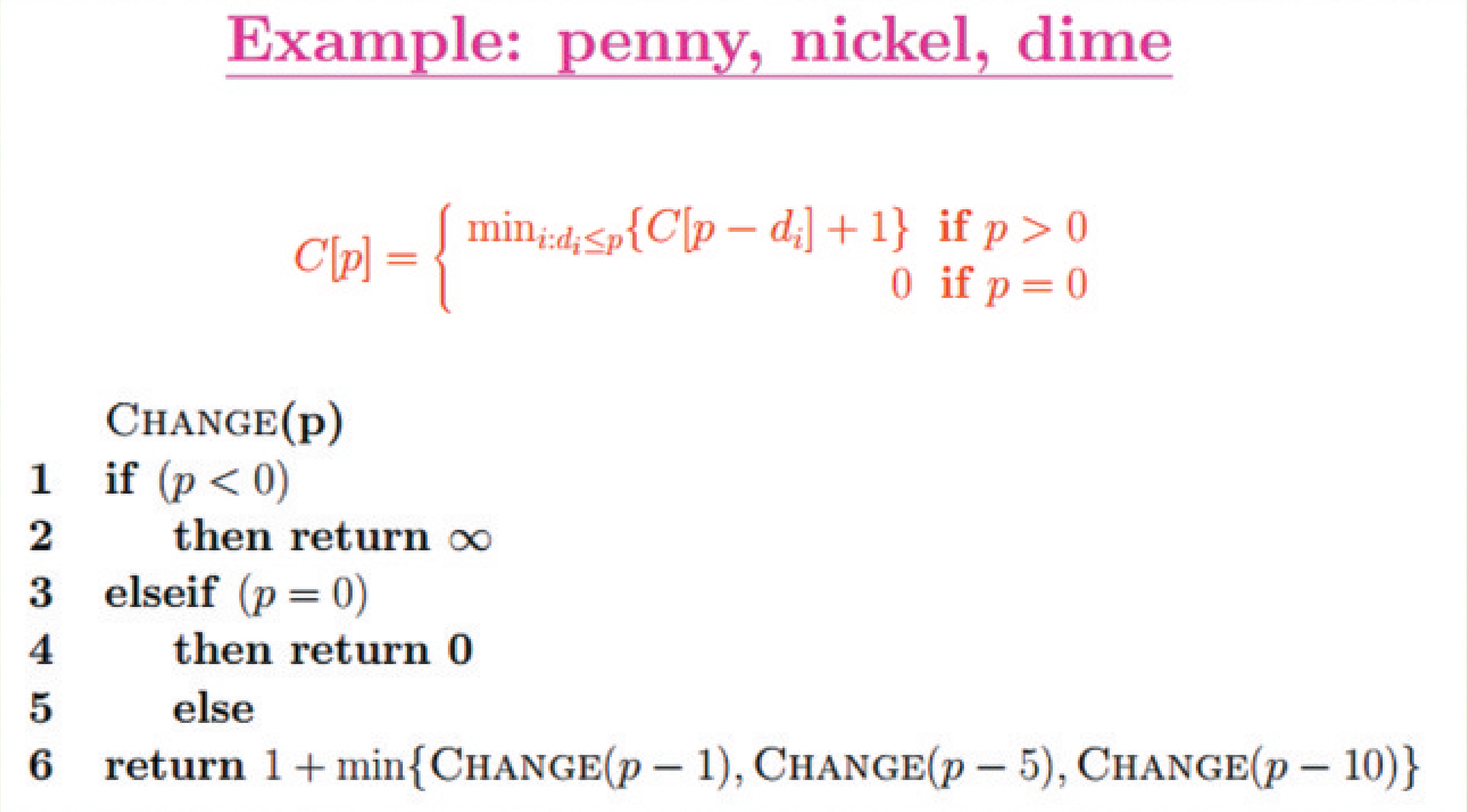
If we have Rs 36 , than coin change algo works like follow:





Solutions

- Sort coins from large to small based on amount
  - 1. Determine how many large coins used
  - 2. Calculate how much the amount left using a given number of large coins
  - 3. Recursion to the deducted value using sub-set of the coins
  - 4. Print out of the remaining amount dividable by the smallest coin amount



Example: penny, nickel, dime

$$C[p] = \begin{cases} \min_{i:d_i \le p}\{C[p-d_i]+1\} & \textbf{if } p > 0 \\ 0 & \textbf{if } p = 0 \end{cases}$$

```
CHANGE(p)
if ($p < 0$)
    then return $\infty$
elseif ($p = 0$)
    then return 0
    else
return $1 + \min\{$CHANGE$(p-1)$, CHANGE$(p-5)$, CHANGE$(p-10)\}$
```

35€!!

Group Members

Abdul Rauf (15B-089-SE) and Moazam Nisar (15B-088-SE)