Problem31

May 5, 2022

1 Problem 31

1.1 Coin Sums

In the United Kingdom the currency is made up of pound (£) and pence (p). There are eight coins in general circulation:

`1p, 2p, 5p, 10p, 20p, 50p, £1 (100p), and £2 (200p).`

It is possible to make £2 in the following way:

`1×£1 + 1×50p + 2×20p + 1×5p + 1×2p + 3×1p`

How many different ways can £2 be made using any number of coins?

Just try a straightforward brute force algorithm first - can be used to check later attempts. Trying every combination…

[ ]:

```
c1 = c2 = c5 = c10 = c20 = c50 = c100 = c200 = 0
sum = 0
count = 0
target = 200

def calcSum():
    global c1, c2, c5, c10, c20, c50, c100, c200
    thisSum = 0
    thisSum += c1
    thisSum += c2 * 2
    thisSum += c5 * 5
    thisSum += c10 * 10
    thisSum += c20 * 20
    thisSum += c50 * 50
    thisSum += c100 * 100
    thisSum += c200 * 200

    return thisSum


while sum < target: #£2
    while sum < target: #£1
        while sum < target: #50p
```

```
            while sum < target: #20p
                while sum < target: #10p
                    while sum < target: #5p
                        while sum < target: #2p
                            while sum < target: #1p
                                c1 += 1
                                sum = calcSum()
                                if sum == target: count += 1
                            c1 = 0
                            c2 += 1
                            sum = calcSum()
                            if sum == target: count += 1
                        c1 = c2 = 0
                        c5 += 1
                        sum = calcSum()
                        if sum == target: count += 1
                    c1 = c2 = c5 = 0
                    c10 += 1
                    sum = calcSum()
                    if sum == target: count += 1
                c1 = c2 = c5 = c10 = 0
                c20 += 1
                sum = calcSum()
                if sum == target: count += 1
            c1 = c2 = c5 = c10 = c20 = 0
            c50 += 1
            sum = calcSum()
            if sum == target: count += 1
        c1 = c2 = c5 = c10 = c20 = c50 = 0
        c100 += 1
        sum = calcSum()
        if sum == target: count += 1
    c1 = c2 = c5 = c10 = c20 = c50 = c100 = 0
    c200 += 1
    sum = calcSum()
    if sum == target: count += 1

print(count)
```

```
73682
```

11 seconds - Should be able to speed this up with more efficient algorithm.

For each possible value up to the target value (£2 in this case) add up the number of combinations due to each individual coin, starting from the lowest. So each value (1,2,3,4,5 … 200) has one way of being made from 1p coins. Then values 2+ have additional combinations due to the 2p, 2 has 2 combinations (1p,1p or 2p), 4 has 3 arrangements (1 with all 1ps and 2 which involve the 2p coin)

[ ]:

```python
coins = [1, 2, 5, 10, 20, 50, 100, 200]
target = 200
coin_arrangements = [0] * (target + 1)
coin_arrangements[0] = 1

for coin in coins:
    for n in range(coin, target + 1):
        coin_arrangements[n] += coin_arrangements[n - coin]

print(coin_arrangements[target])
```

```
73682
```

0.1seconds - at least 3 orders of magnitude faster