|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Original code | | Optimize code | |  |
| Size Amount | Coins | Run Time | Size Amount | Coins | Runtime (s) |
| 65 | 1, 5 | 119.5 | 65 | 1, 5 | 0.00341 |
| 65 | 1, 5, 10 | 473 | 65 | 1, 5, 10 | 0.00203 |
| 65 | 1, 5, 10, 25 | 674.38 | 65 | 1, 5, 10, 25 | 0.00112 |
| 65 | 1, 5, 8, 10, 25 | Too long | 65 | 1, 5, 8, 10, 25 | 0.00194 |
| 65 | 1, 5, 8, 10, 15, 25 | Too long | 65 | 1, 5, 8, 10, 15, 25 | 0.00299 |
| 65 | 1, 5, 7, 8, 10, 12, 15, 25 | Too long | 65 | 1, 5, 7, 8, 10, 12, 15, 25 | 0.00498 |
| 65 | 1, 5, 7, 8, 10, 12, 15, 20, 25 | Too long | 65 | 1, 5, 7, 8, 10, 12, 15, 20, 25 | 0.000998 |

My updated programming algorithm to solve the change-making problem. Num\_coins take three parameters: the amount of change we want to make, a list of valid coin values, and the minim number of coins for each value.

For my updated code, in this loop in line 6, we consider using all possible coins to make change for the amount specified by cents. Also, it keeps track of the coins used by remembering the last coin we add for each entry. If I know the last coin which were added, my code can subtract the value of the coin. Then we get the previous amount. We can keep tracing back until we get to the beginning.