Problem 31 - Coin Sums

Gautam Manohar

24 June 2018

This document originally appeared as a blog post on my website. Find it at `gautammanohar.com/euler/31`.

1 Problem Statement

In England, the currency is made up of pounds (£) and pence (p), and 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:

$$1 \cdot £1 + 1 \cdot 50\text{p} + 2 \cdot 20\text{p} + 1 \cdot 5\text{p} + 1 \cdot 2\text{p} + 3 \cdot 1\text{p}$$

In how many ways can N pence be made from any denomination of coins? Output your answer modulo $10^9 + 7$.

2 My Algorithm

This problem is a classic application of dynamic programming, the technique of breaking up a problem into smaller, reusable chunks. To make N pence, we can first make $N - c$ pence, then add one c pence coin, where c is the value of some denomination. Then the number of ways we can make N pence is the sum of $N - c$ over all valid c.

Using this principle, our solution is simple. We maintain a list `coins` such that `coins[n]` is the number of ways we can make n pence, where $0 \leq n \leq N$. Then, for each valid denomination of coins c, we loop over $c \leq n \leq N$ and increment `coins[n]` by `coins[n-c]`. At each step, we perform addition modulo $10^9 + 7$.

Because we maintain the list for all $0 \leq n \leq N_{\max}$, we can answer each query in $O(1)$ time. And so our solution has time complexity $O(CN_{\max} + T)$, where C is the number of coins and T is the number of queries.