

You have a set of n coins in a bag, each coin having a value between 1 and m , where $m \geq n$. Some coins might have the same value. You pick two coins at random, with replacement, and record the sum of their values. In other words, once you record the value of the first coin, you immediately place it back into the bag and pick up another coin.

Describe an $O(m \log m)$ algorithm to return the set of possible sums that can be achieved.

Note. You know the set of coin values in the bag ahead of time.

Hint. Use FFT.

Rubric.

- This task will form part of the portfolio.
- Ensure that your argument is clear and keep reworking your solutions until your lab demonstrator is happy with your work.

Solution. We begin by initialising an array of size m representing the count of each value, initialised all as zero. Given that we know the coins in the bag, we can easily assign the counts of each value on the coins to their corresponding indices in the array. This can be done in $O(m)$ time. Next, we can use the FFT to perform a convolution to create a new array, which calculates the possible ways to reach the index value from a sum of two of the given values. This is done in $O(m \log m)$ time. Finally, we simply count the number of values in the result that aren't zero and return that value, done in $O(2m) = O(m)$ time. This algorithm is therefore $O(m \log m)$ time.