

# Enumerating all the ways of making change

 [devblogs.microsoft.com/oldnewthing/20140915-00](http://devblogs.microsoft.com/oldnewthing/20140915-00)

September 15, 2014



Raymond Chen

Today's Little Program enumerates all the ways of making change for a particular amount given a set of available denominations.

The algorithm is straightforward. To make change for a specific amount from a set of available denominations, you can take one denomination and decide how many of those you want to use. Then use the remaining denominations to make change for the remainder.

For example, if the available coins have values [1, 5, 10, 25] and you want to make change for 60 cents, you first decide how many 25-cent pieces you want to use. If you use none, then you need to make change for 60 cents using just [1, 5, 10]. If you use one, then you need to make change for 35 cents using just [1, 5, 10]. And if you use two, then you need to make change for 10 cents using just [1, 5, 10].

(We use the largest coin first to reduce the number of dead ends, like asking "Please make change for 83 cents using only 10-cent coins.")

```
function MakeChange(coins, total, f) {
  if (total == 0) { f([]); return; }
  if (coins.length == 0) return;
  var coin = coins[coins.length - 1];
  var remaining = coins.slice(0, -1);
  var used = [];
  for (var target = total; target >= 0; target -= coin) {
    MakeChange(remaining, target, function(s) {
      f(used.concat(s));
    });
    used.push(coin);
  }
}
```

```
MakeChange([1, 5, 10, 25], 60, console.log.bind(console));
```

First, we take care of some base cases. To make change for zero cents, we simply use zero coins. And it's not possible to make change for a nonzero amount with no coins.

Otherwise, we take the highest denomination coin and try using zero of them, then one of them, and so on, until we exceed the total amount necessary.

There are related problems, such as determining whether a particular amount of change can even be made, given a collection of denominations and calculating the number of ways change can be made rather than enumerating them. But I like enumeration problems.

Raymond Chen

**Follow**

