The Coin Row Problem

Given n coins in row with values $C_1, C_2, \ldots C_n$ we want to pick up coins to maximize the amount of money, subject to the constraint that two adjacent coins in the initial row may not be picked up.

The Recurrence

Let $F(n)$ be the maximum amount that can picked up from a row of n coins. Then

$$F(n) = \begin{cases} 0 & \text{if } n = 0 \\ C_1 & \text{if } n > 1 \\ \max\{c_n + F(n-2), F(n-1)\} & \text{if } n > 1 \end{cases}$$

The Algorithm

```
CoinRow(C[1 .. n])
    F[0] = 0
    F[1] = C[1]

    for i = 2 to n
        F[i] = max(C[i] + F[i-2], F[i-1])

    return F[n]
```

Recovering the Solution

The above algorithm gives the *value* of the optimal solution, but it does not actually say which coins were used to construct that solution. We will need to enhance the procedure so that once and optimal solution is found, we can trace backwards to see which coins contributed to that solution.