1 Linear homogeneous recurrence relations with constant coefficients

Linear homogeneous recurrence relations with constant coefficients A linear homogeneous recurrence relations with constant coefficients of order k is a sequence with the following recursive relation

$$a_n = \sum_{i=1}^{k} c_i a_{n-i}$$

, where c_i are constants.

We use linear recurrence relation to abbreviate.

The most common example is the Fibonacci sequence. $a_0 = 0, a_1 = 1$ and $a_n = a_{n-1} + a_{n-2}$. The Fibonacci sequence have a simple implementation. $fibs = 0 : 1 : zipWith\ (+)\ fibs\ (tail\ fibs)$. We want to generalize it.

1.1 Lazy sequence

Problem 1. *Input*:

1. A list of coefficients $[c_1, c_2, \ldots, c_n]$ of a linear recurrence relation.
2. A list of base cases $[a_0, a_1, \ldots, a_{n-1}]$ of a linear recurrence relation.

Output: The sequence of values of the linear recurrence relation as a infinite list $[a_0, a_1, \ldots$.

```
import Data.List
linearRecurrence :: Num a => [a] -> [a] -> [a]
linearRecurrence coef base = a
  where a = base ++ map (sum . (zipWith (*) coef)) (map (take n) (tails a))
        n = (length coef)
```

Having a infinite list allows simple manipulations. However, finding the nth number in the sequence cost $O(nk)$ time. It becomes unreasonable if a person only need to know the nth number.

1.2 Determine nth element in the index

If n is very large, a more common technique would be solve for a_n using matrix multiplication.