Fibonacci coding

Fibonacci series

0 1 1 2 3 5 8 13 ……

The next number is the sum of previous two numbers in the series

Fibonacci coding

Encodes a number into binary (0 and 1) using the fibonacci representation of the number

Zeckendorf’ theorem

Every positive number can be uniquely represented as the sum of distinct non-neighbouring fibonacci numbers

Example

Suppose we start with n=**143**. The first **f** will be **89**. We mark it as used:

Fibonaccis	1	2	3	5	8	13	21	34	55	**89**	144
Usage bit	0	0	0	0	0	0	0	0	0	**1**	-

Now **n** = 143 - 89 = **54**. Fibonacci in hand is 55 which is > than 54. We mark it unused:

Fibonaccis	1	2	3	5	8	13	21	34	**55**	89	144
Usage bit	0	0	0	0	0	0	0	0	**0**	1	-

n = 54. f = 34. We mark it as used:

Fibonaccis	1	2	3	5	8	13	21	**34**	55	89	144
Usage bit	0	0	0	0	0	0	0	**1**	0	1	-

And finally to n = **0:**

Fibonaccis	1	2	3	5	8	13	21	34	55	89	144
Usage bit	0	1	0	1	0	1	0	1	0	1	-

For the codeword, read the second row of above table from left to right: 0101010101
Append additional '1' bit: 0101010101**1**
Final codeword for 143 = 01010101011

Algorithm

- Take 24 as an example
- Use the greedy way to find the nearest possible fibonacci number 21
- Subtract it from 24 (24-21 = 3) and repeat the same for 3, till you don’t reach 0

Find the closest fibonacci number

- If it is 0 or 1 return 0 or 1 respectively. That itself is the closest number
- Initialize f1,f2,f3 as 0,1,1
- Till f3 is less than n (20 and 3 respectively in our case)
- f1=f2 and f2 = f3 and f3=f1+f2
- First time you get 21 and next time you get 3

Usage

- Since every number can be represented uniquely, it can be used as an alternate method of representation of a number, like binary
- The number of 1’s is lesser than binary
- Used in data compression techniques where the 0 bits can be ignored and only the 1’s can be sent