## Arithmetic of Fibonacci representations

I.S. Sergeev, 2021

Fibonacci codes:

no '11' subwords

The number of length-
$$n$$
 words ...  $\{\bigstar^n\} = \{0\bigstar^{n-1}\} \cup \{10\bigstar^{n-2}\}$ 

... is 
$$\Phi_{n+2}$$
 ( $\Phi_k$  — Fibonacci numbers:  $\Phi_1 = \Phi_2 = 1$ ,  $\Phi_k = \Phi_{k-1} + \Phi_{k-2}$ ).

$$0=0000,\ 1=0001,\ 2=0010,\ 3=0100,\ 4=0101,\ 5=1000,\ 6=1001,\ 7=1010$$

Fibonacci (canonical) representation of a number  $A: [a_n, \ldots, a_3, a_2]$ :

$$A = \sum_{k=0}^{n} a_k \Phi_k, \qquad a_k \in \{0, 1\}, \qquad a_{k+1} a_k = 0.$$

Lines of research:

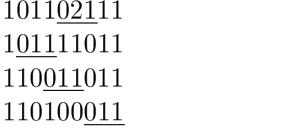
(i) arithmetic operations in Fibonacci encoding (+, -, \*)

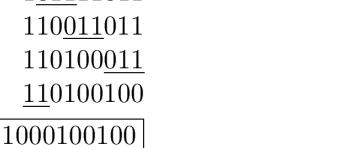
(ii) transitions between the Fibonacci and the binary representations of a number

Addition (subtraction) of Fibonacci encodings:

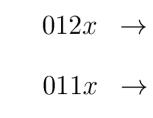
Complexity O(n)

$$\begin{array}{rcl}
+ & 10101010 \\
& & 10100101 \\
& & \underline{20}201111 \\
& & 100301111 \\
& & & 101102111
\end{array}$$









Rules:

$$\begin{array}{ccc}
021x & \to & 110x \\
012x & \to & 101x
\end{array}$$

 $030x \rightarrow 110x'$ 

 $020x \rightarrow$ 



100x'

[Berstel, Frougny, Ahlbach, Usatine, Pippenger ≈1986–2011]

Multiplication (???)

in binary encoding:  $M(n) = O(n \log n)$ [Harvey, van der Hoeven 2019] Transition to binary encoding:  $O(n^2)$ [Ahlbach, Usatine, Pippenger 2011]

Schönhage's method: 
$$A = [a_{k-1}, \dots, a_0]_b \longrightarrow A = [a'_{n-1}, \dots, a'_0]_2$$

$$A = [A_1]_b \cdot b^{k/2} + [A_0]_b$$
 Complexity:

$$\downarrow \qquad T(n) \le 2T(n/2) + M(n/2) + O(n)$$

$$A = [A_1]_2 \cdot [b^{k/2}]_2 + [A_0]_2 \Longrightarrow T(n) = O(M(n) \log n)$$

Lucas numbers: 
$$L_m = \Phi_{m+1} + \Phi_{m-1} = \varphi_+^m + \varphi_-^m$$
, where  $\varphi_{\pm} = \frac{1 \pm \sqrt{5}}{2}$   
 $\Phi_k L_m = \Phi_{m+k} - (-1)^k \Phi_{m-k}$  (\*)

Algorithm: 
$$A = [a_{n-1}, \dots, a_0]_2 \longrightarrow A = [a'_{k-1}, \dots, a'_0]_{\varphi}$$

$$A = A_1 \cdot L_m + A_0$$
 division with remainder by  $L_m \approx 2^{n/2}$ 

$$A = [A_1]_{\varphi} \cdot L_m + [A_0]_{\varphi} \stackrel{(\star)}{=} [A^+]_{\varphi} - [A^-]_{\varphi} + [A_0]_{\varphi}$$

Complexity: 
$$T(n) \le 2T(n/2) + D(n) + O(n) \Rightarrow T(n) = O(M(n) \log n)$$

r-Fibonacci representations: <u>no r ones in a row</u>

r-bonacci numbers: 
$$\Phi_n^{(r)} = \Phi_{n-1}^{(r)} + \ldots + \Phi_{n-r}^{(r)}$$

$$A = \sum_{k=2}^{n} a_k \Phi_k^{(r)}, \qquad a_k \in \{0, 1\}, \qquad a_{k+1} \cdot \dots \cdot a_{k+r} = 0.$$

 $\langle q \rangle$ -Fibonacci representations: ones are separated by  $\geq q-1$  zeros

$$\langle q \rangle$$
-bonacci numbers:  $\Phi_n^{\langle q \rangle} = \Phi_{n-1}^{\langle q \rangle} + \Phi_{n-q}^{\langle q \rangle}$ 

$$A = \sum_{k=0}^{n} a_k \Phi_k^{\langle q \rangle}, \qquad a_k \in \{0, 1\}, \qquad a_{k+1} + \ldots + a_{k+q} \le 1.$$

transition  $[A]_2 \longleftrightarrow [A]_{\varphi(r)}, \quad [A]_2 \longleftrightarrow [A]_{\varphi(3)}$ :  $T(n) = 2^{O(\sqrt{\log n})} n.$ 

Addition of Tribonacci encodings: O(n).

Addition/subtraction in other cases: (???).

## References:

Ahlbach C., Usatine J., Frougny C., Pippenger N. Efficient algorithms for Zeckendorf arithmetic // Fibonacci Quarterly. 2013. 51(3), 249–255.

Sergeev I.S. On the complexity of Fibonacci coding // Problems of Information Transmission. 2018. 54(4), 343-350.