P47

Restrictions:

$1 \leq n \leq 10^{30}$

"A" $\rightarrow$ "AB"

"B" $\rightarrow$ "A"

Example:

A	(1)	$A(1) = 1$	$B(1) = 0$
AB	(2)	$A(2) = 1$	$B(2) = 1$
ABA	(3)	$A(3) = 2$	$B(3) = 1$
ABAAB	(4)	$A(4) = 3$	$B(4) = 2$
ABAABABA	(5)	$A(5) = 5$	$B(5) = 3$
...	(...)	...	...

N° of occurrences of the letter A after the n^{th} iterations

N° of occurrences of the letter B after n^{th} iteration

$T(n) \leftarrow$ Length of the sequence after the n^{th} iteration

$$\boxed{T(n) = A(n) + B(n)}$$

$T(1) = 1$

$T(2) = 2$

$T(3) = 3$

$T(4) = 5$

$T(5) = 8$

$T(6) = 13$

...

After some observation, I concluded that:

$B(n+1) = A(n)$

$A(n+1) = A(n) + B(n)$

$\therefore T(n+1) = A(n+1) + B(n+1)$

$= A(n) + B(n) + A(n) =$

$= 2 \times A(n) + B(n)$

Expansion rate of the sequence

$T(n+1) - T(n) = R \Leftrightarrow$

$\Leftrightarrow 2 \times A(n) + B(n) - (A(n) + B(n)) = R \Leftrightarrow$

$\Leftrightarrow \cancel{2} \times A(n) + \cancel{B(n)} - \cancel{A(n)} - \cancel{B(n)} = R \Leftrightarrow$

$\Leftrightarrow A(n) = R \Rightarrow T(n+1) - T(n) = A(n)$

So, we know that

$T(n+1) - T(n) = A(n) \quad (1)$

and we also know that

$A(n+1) = A(n) + B(n)$

which is the same as

$A(n) = A(n-1) + B(n-1) \quad (2)$

by combining (1) and (2)

$$T(n+1) - T(n) = A(n) \Leftrightarrow$$

$$\Leftrightarrow T(n+1) - T(n) = \underbrace{A(n-1) + B(n-1)}_{= T(n-1)} \Leftrightarrow$$

$$\Leftrightarrow T(n+1) - T(n) = T(n-1) \Leftrightarrow$$

$$\Leftrightarrow T(n+1) = T(n) + T(n-1)$$

This expression is the equivalent of

$$\boxed{T(n) = T(n-1) + T(n-2)}$$

by considering that $T(0) = 1$ and $T(1) = 1$, our succession $T(n)$ behoves just as <u>Fibonacci Sequence</u>.

Now, we can view our sequence from a different perspective.

A (1)

AB (2)

ABA (3)

ABA AB (4)

ABAAB ABA (5)

$\underbrace{ABAAB}_{SEQ(4)}\ \underbrace{ABA}_{SEQ(3)}$

... (...)

Imagine the sequence as a block of lego: we can decomposite it until it's only constituted of the blocks A and AB

$$SEQ(5) = SEQ(4) + SEQ(3) \iff$$

$$\iff SEQ(5) = (SEQ(3) + SEQ(2)) + (SEQ(2) + SEQ(1))$$

$$\iff \boxed{SEQ(5) = SEQ(2) + SEQ(1) + SEQ(2) + SEQ(2) + SEQ(1)}$$

The function SEQ(n) represents the ordered sequence of letters of the n^{th} iteration

Algorithm Design

With this in mind, we can design an algorithm that decomposes our SEQ(i) into blocks of SEQ(2) and SEQ(1), where i represents the number of iterations that first satisfies the condition, given an n:

$$\boxed{n \leq T(i)}$$

To demonstrate how the algorithm works, consider the following example:

(1) $\boxed{A}$ $T(1) = 1$ $n = 11$

(2) $\boxed{AB}$ $T(2) = 2$

(3) ABA $T(3) = 3$

(4) ABAAB $T(4) = 5$

(5) ABAABABA $T(5) = 8$

(6) ABAABABAABAAB $T(6) = 13$

Since $i = 6$ is the first number that satisfies the condition $n \leq T(i)$, our sequence will be SEQ(6).

Lets decompose SEQ(6) into 2 blocks.

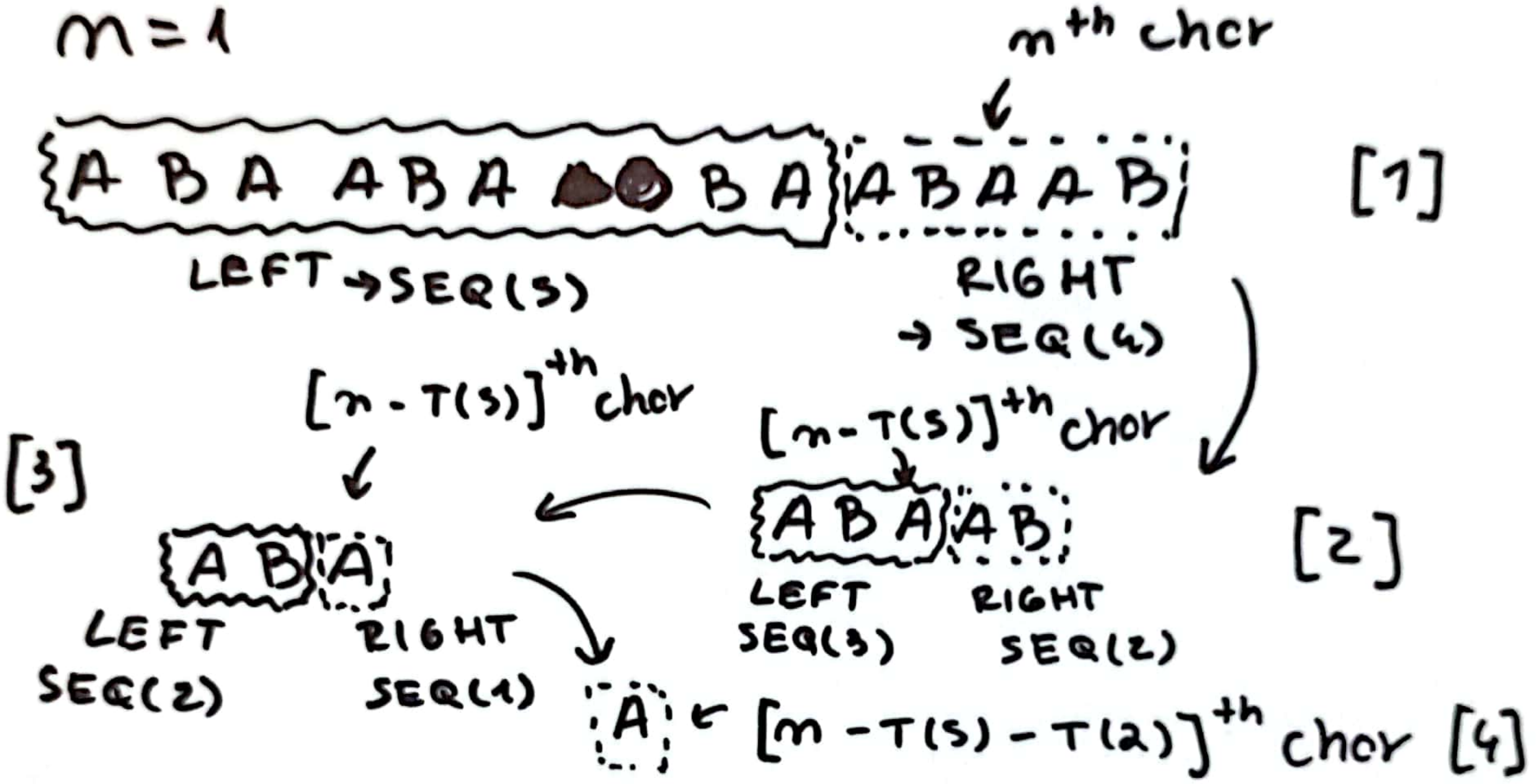
$$\underbrace{\boxed{ABAABABA}}_{SEQ(5)}\underbrace{\boxed{ABAAB}}_{SEQ(4)}$$

Since we proved that every block where $T(b) > 2$ can be formed by SEQ(1) and SEQ(2), it's intuitive that, for every division step, our n^{th} char will be part of the left or right block

SEQ(n-1) or SEQ(n-2)

Thus, for every division process, will select the brench where our n falls on.

$n = 1$



In step [3] our sequence was fully decomposed and when selecting the right side (SEQ(1)) we end up with the n^{th} char of the SEQ(6), the equivalent of the $[n - T(5) - T(2)]^{th}$ char of the SEQ(1)

↳ $11 - 8 - 2 = 1^{th}$ which is A.

Algorithm:

1° - Receive n, the n^{th} char of the sequence

2° - Discover i that first satisfies the condition: $n \leq T(i)$, where $T(i)$ corresponds to the length of the sequence after the i^{th} iteration

3° - Determine your initial sequence as $SEQ(i)$, set $k = n$, initially, and set $p = i$

4° - while $p > 2$, cycle through:

↳ 4.1° - Decompose $SEQ(p)$ as $SEQ(p-1)$ and $SEQ(p-2)$

↳ 4.2° - verifies where k falls on: left ($k \leq T(p-1)$) or right ($k > T(p-1)$)

↳ 4.3° - IF falls on left, set $p = p-1$ and repeat the cycle starting at 4.1°

↳ 4.4° - IF falls on right, set $p = p-2$, update $k = k - T(p-1)$ and repeat the cycle at 4.1°

5° - Return the k^{th} char at the SEQ(p)

Check the script at

github.com/mmroch4

Written by:

Miguel Martins Rocha