

I-ésimo Término de Tribonacci (Problema 7)

March 29, 2024

1 Descripción del Ejercicio

La sucesión recurrente de Tribonacci se define por la función $T(n)$, con $n \geq 0$ donde:

- $F(0) = 0$
- $F(1) = 0$
- $F(2) = 1$
- $F(n) = F(n-1) + F(n-2) + F(n-3)$

Dado un número i , hallar el i -ésimo término de la sucesión Tribonacci.

Salida

Imprimir el i -ésimo término de Tribonacci, es decir, $T(i)$.

Por ejemplo: Para $i = 6$

Debería imprimir:

7

Logisim

Se dispondrá en *INPUT* los datos de entrada a partir de la dirección 0. La entrada se estructura de la siguiente forma:

- $w_0 : i$

SASM

En la sección `.data` se deben definir los valores de entrada de la siguiente forma:

```
i dw 6
```

2 Pseudocódigo

```
1 def Tribonacci(n):
2     b = 0
3     c = 0
4     a = 1
5     while ( n > 2):
6         a = a+b+c
7         c, b = b, c
8         b, a = a, b
9         n = n-1
10    if ( a <= 1)
11        a = 0
12    return a
13
14
```

Listing 1: Función Tribonacci en Python

3 Código en Ensamblador

```
1 %include "io.inc"
2
3 section .data
4 i dw 6
5
6 section .text
7 global main
8 main:
9 mov ebp, esp; for correct debugging
10 mov eax, 1 ; valor mas actual f(n-1)
11 mov ebx, 0 ; valor anterior f(n-2)
12 mov edx, 0 ; valor mas anterior f(n-3)
13
14 xor ecx, ecx ; inicializa en 0 ecx
15
16 movzx ecx, word [i] ; guarda el valor de i en ecx
17 ;mov ecx, i
18
19 ; Aqui se definen los casos bases si ecx es 0 o 1 => 0, si ecx
20 es 2=>1
21 cmp ecx, 1 ; compara ecx con 1
22 jle .base ; si es menor igual que 1
23
24 cmp ecx, 2 ; compara ecx con 2
25 je .end ;si es igual a 2
26
27 ; Si es mayor que 2, es decir no entra en los casos bases
28 sub ecx, 2 ; resto 2 para quedarme con la cantidad de veces que
29 debo ejecutar el codigo
30 jmp .trib ; entra al bloque de codigo del tribonacci
31
32 .base:
```

```

32     mov eax, 0
33     jmp .end
34
35     .trib:
36     add edx, eax ; sumar edx con eax y guardarlo en edx
37     add edx, ebx ; sumar ebx con edx y guardarlo en edx
38
39     ; Esto permite dejar el valor de la ultima suma en orden de
    prioridad y eliminar el valor anterior
40     xchg edx, ebx ; intercambiar los valores edx y ebx
41     xchg ebx, eax ; intercambiar los valores ebx y eax
42
43     ; Decrementar el contador de las veces que debia ejecutar el
    codigo y comparar con 0
44     ; si es 0 ya terminamos y sale hasta .end si aun quedan veces
    por ejecutar salta nuevamente
45     ; al inicio del trib a ejecutar otra vez
46     dec ecx
47     cmp ecx, 0
48     je .end
49
50     jmp .trib
51
52     .end:
53     PRINT_DEC 1, eax
54
55     xor eax, eax
56     ret
57

```

Listing 2: Código Tribonacci en Ensamblador

Para guardar el valor de i q tiene una extension de 16 bits en un ecx de 32 bits se utiliza la instrucción movzx que extiende ceros para llenar los bits de ecx.

4 Diagrama de la Máquina de Estado

Para visualizar la máquina de estado, se incluye a continuación el diagrama correspondiente:

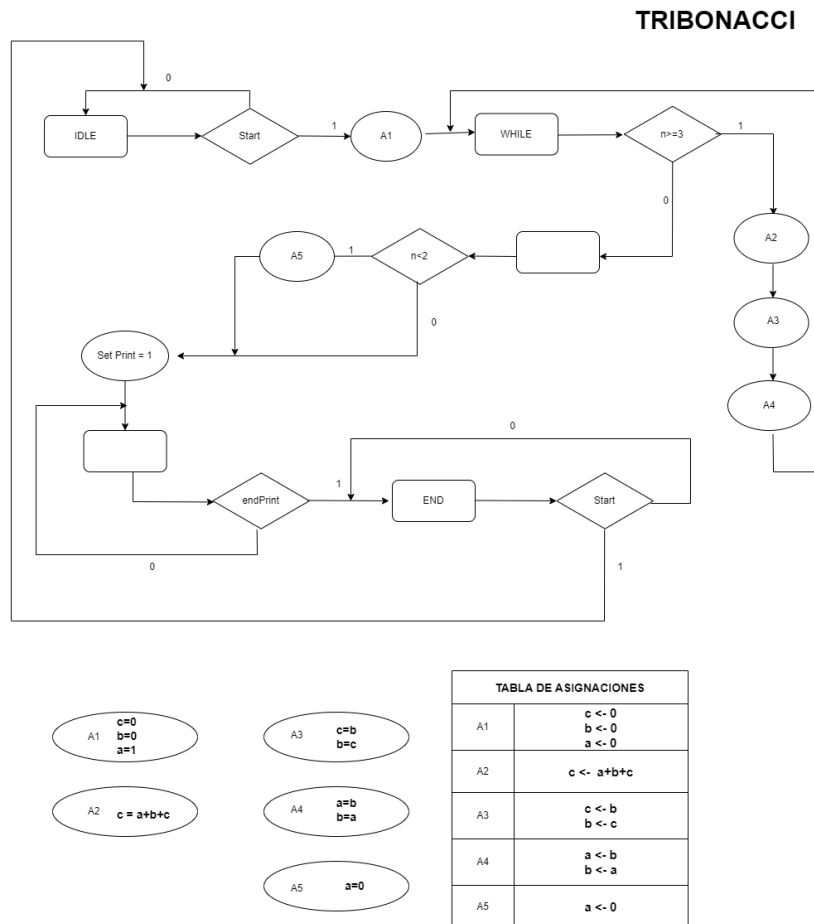


Figure 1: Diagrama de la máquina de estado.