Instituto Politécnico Nacional

Escuela Superior de Cómputo

ANÁLISIS DE ALGORITMOS RECURSIVOS

Ejercicio 04

Análisis de Algoritmos
M. en C. Edgardo Adrián Franco Martínez
Grupo: 3CM3
Fecha: 08 / Mayo / 2018

Alumno:

Calva Hernández José Manuel 2017630201

Índice

Algoritmo 1

```
int FuncionRecursiva(int num )
{
		if ( num == 0 )
				return 1;
		else if ( num < 3 )
		{
				resultado=0;
				for(i=0;i<num*num;i++)
						resultado*=num;
		}		return resultado;
		else
				return FuncionRecursiva( num - 1 )*FuncionRecursiva(num - 2)/FuncionRecursiva(num - 3);
}
```

$$T(0) = 0$$

$$T(1) = 1\ (multiplicación\ dentro\ de\ la\ iteración)$$

$$T(2) = 4\ (multiplicaciones\ dentro\ de\ la\ iteración)$$

$$T(n) = T(n-1) + T(n-2) + T(n-3) + 2\ (multiplicación\ y\ división)$$

$$T(n) - T(n-1) - T(n-2) - T(n-3) = 2$$

$$(x^3 - x^2 - x - 1)(x - 2) = 0$$

$$r_1 \approx 1.8393\ , r_2 \approx -0.41964 - 0.60629i\ , r_3 \approx -0.41964 + 0.60629i, r_4 = 2$$

$$T(n) \approx c_1(1.8393)^n + c_2(-0.41964 - 0.60629i)^n + c_3(-0.41964 + 0.60629i)^n + c_4 2^n$$

$$T(3) = T(3-1) + T(3-2) + T(3-3) + 2 = T(2) + T(1) + T(0) + 2 = 4 + 1 + 0 + 2 = 7$$

Sistemas de ecuaciones:

$$T(0) \approx c_1(1.8393)^0 + c_2(-0.41964 - 0.60629i)^0 + c_3(-0.41964 + 0.60629i)^0 + c_4 2^0 = c_1 + c_2 + c_3 + c_4 = 0$$

$$T(1) \approx c_1(1.8393)^1 + c_2(-0.41964 - 0.60629i)^1 + c_3(-0.41964 + 0.60629i)^1 + c_4 2^1 = 1$$

$$T(2) \approx c_1(1.8393)^2 + c_2(-0.41964 - 0.60629i)^2 + c_3(-0.41964 + 0.60629i)^2 + c_4 2^2 = 4$$

$$T(3) \approx c_1(1.8393)^3 + c_2(-0.41964 - 0.60629i)^3 + c_3(-0.41964 + 0.60629i)^3 + c_4 2^3 = 7$$

$$c_1 \approx 1.41964\ ;\ c_2 \approx 0.290178\ +\ 0.303145\mathrm{i}\ ;\ c_3 \approx 0.290178 -\ 0.303145\mathrm{i}\ ;\ c_3 = -1$$

$$\therefore T(n) \approx (1.41964)(1.8393)^n + (0.290178 +\ 0.303145\mathrm{i})(-0.41964 - 0.60629i)^n + (0.290178 -\ 0.303145\mathrm{i})(-0.41964 + 0.60629i)^n - 2^n$$

Por lo tanto, la recurrencia queda como:

$$T(n) = O(1.8393)^n$$

Algoritmo 2

```
int Tribonacci( int num )
{

          if (num==0)
                    return 0;
          else if   (num==1||num==2)
                    return 1;
          else
                    return Tribonacci(num-1)+Tribonacci(num-2)+Tribonacci(num-3);
}
```

$$T(0) = 0\ ;\ T(1) = 1\ ;\ T(2) = 1$$

$$T(n) = T(n-1) + T(n-2) + T(n-3) + 2$$

$$T(n) - T(n-1) - T(n-2) - T(n-3) = 2$$

$$(x^3 - x^2 - x - 1)(x - 2) = 0$$

$$r_1 \approx 1.8393\ , r_2 \approx -0.41964 - 0.60629i\ , r_3 \approx -0.41964 + 0.60629i, r_4 = 2$$

$$T(n) \approx c_1(1.8393)^n + c_2(-0.41964 - 0.60629i)^n + c_3(-0.41964 + 0.60629i)^n + c_4 2^n$$

$$T(3) = T(3-1) + T(3-2) + T(3-3) + 2 = T(2) + T(1) + T(0) + 2 = 1 + 1 + 0 + 2 = 4$$

Sistemas de ecuaciones:

$$T(0) \approx c_1(1.8393)^0 + c_2(-0.41964 - 0.60629i)^0 + c_3(-0.41964 + 0.60629i)^0 + c_4 2^0 = c_1 + c_2 + c_3 + c_4 = 0$$

$$T(1) \approx c_1(1.8393)^1 + c_2(-0.41964 - 0.60629i)^1 + c_3(-0.41964 + 0.60629i)^1 + c_4 2^1 = 1$$

$$T(2) \approx c_1(1.8393)^2 + c_2(-0.41964 - 0.60629i)^2 + c_3(-0.41964 + 0.60629i)^2 + c_4 2^2 = 1$$

$$T(3) \approx c_1(1.8393)^3 + c_2(-0.41964 - 0.60629i)^3 + c_3(-0.41964 + 0.60629i)^3 + c_4 2^3 = 4$$

$$c_1 \approx 0.871233;\ c_2 \approx 0.564384\ -\ 0.718494\mathrm{i}\ ;\ c_3 \approx 0.564384 +\ 0.718494\mathrm{i}\ ;\ c_3 = -1$$

$$\therefore T(n) \approx (0.871233)(1.8393)^n + (0.564384\ -\ 0.718494\mathrm{i})(-0.41964 - 0.60629i)^n + (0.564384 +\ 0.718494\mathrm{i})(-0.41964 + 0.60629i)^n - 2^n$$

Por lo tanto, la recurrencia queda como:

$$T(n) = O(1.8393)^n$$

Algoritmo 3

$$T(n) = 3T(n-1) + 4T(n-2) \Rightarrow n > 1\,; T(0) = 0\,; T(1) = 1$$

$$T(n) - 3T(n-1) - 4T(n-2) = 0$$

$$x^2 - 3x - 4 = 0$$

$$(x-4)(x+1) = 0$$

$$r_1 = 4\,, r_2 = -1$$

$$T(n) = c_1 4^n + c_2(-1)^n$$

Sistema de ecuaciones:

$$T(0) = c_1 4^0 + c_2(-1)^0 = c_1 + c_2 = 0$$

$$T(1) = c_1 4^1 + c_2(-1)^1 = 4c_1 - c_2 = 1$$

$$c_1 = \frac{1}{5}\ ;\ c_2 = -\frac{1}{5}$$

$$\therefore T(n) = \frac{1}{5}4^n - \frac{1}{5}(-1)^n$$

Por lo tanto, la recurrencia queda como:

$$T(n) = O(4^n)$$

Algoritmo 4

$$T(n) = 3T(n-1) + 4T(n-2) + (n+5)2^n \Rightarrow n > 1\,; T(0) = 5\,; T(1) = 27$$

$$T(n) - 3T(n-1) - 4T(n-2) = (n+5)2^n$$

$$(x^2 - 3x - 4)(x-2)^2 = 0$$

$$(x-4)(x+1)(x-2)^2 = 0$$

$$r_1 = 4\,, r_2 = -1\,, r_3 = 2\,, r_4 = 2$$

$$T(n) = c_1 4^n + c_2(-1)^n + c_3 2^n + c_4 n 2^n$$

$$T(2) = 3T(2-1) + 4T(2-2) + (2+5)2^2 = 3T(1) + 4T(0) + (7)(4) = 3(27) + 4(5) + 28 = 81 + 20 + 28 = 129$$

$$T(3) = 3T(3-1) + 4T(3-2) + (3+5)2^3 = 3T(2) + 4T(1) + (8)(8) = 3(129) + 4(27) + 64 = 387 + 108 + 64 = 559$$

Sistema de ecuaciones:

$$T(0) = c_1 4^0 + c_2(-1)^0 + c_3 2^0 + c_4(0)2^0 = c_1 + c_2 + c_3 = 5$$

$$T(1) = c_1 4^1 + c_2(-1)^1 + c_3 2^1 + c_4(1)2^1 = 4c_1 - c_2 + 2c_3 + 2c_4 = 27$$

$$T(2) = c_1 4^2 + c_2(-1)^2 + c_3 2^2 + c_4(2)2^2 = 16c_1 + c_2 + 4c_3 + 8c_4 = 129$$

$$T(3) = c_1 4^3 + c_2(-1)^3 + c_3 2^3 + c_4(3)2^3 = 64c_1 - c_2 + 8c_3 + 24c_4 = 559$$

$$c_1 = \frac{48}{5}\ ;\ c_2 = \frac{57}{5}\ ;\ c_3 = -16\ ;\ c_3 = \frac{7}{2}$$

$$\therefore T(n) = \frac{48}{5}4^n + \frac{57}{5}(-1)^n - (16)2^n + \frac{7}{2}n2^n$$

Por lo tanto, la recurrencia queda como:

$$T(n) = O(4^n)$$

Algoritmo 5

$$T(n) - 2T(n-1) = 3^n \Rightarrow n \geq 2\,; T(0) = 0\,; T(1) = 1$$

$$T(n) - 2T(n-1) = 3^n$$

$$(x-2)(x-3) = 0$$

$$r_1 = 2\,, r_2 = 3$$

$$T(n) = c_1 2^n + c_2 3^n$$

Sistema de ecuaciones:

$$T(0) = c_1 2^0 + c_2 3^0 = c_1 + c_2 = 0$$

$$T(1) = c_1 2^1 + c_2 3^1 = 2c_1 + 3c_2 = 1$$

$$c_1 = -1\ ;\ c_2 = 1$$

$$\therefore T(n) = -2^n + 3^n$$

Por lo tanto, la recurrencia queda como:

$$T(n) = O(3^n)$$

Algoritmo 6

```
int BusquedaBinaria(int num_buscado, int numeros[], int inicio, int centro, int final)
{
            if (inicio>final)
                        return -1;
            else if (num_buscado == numeros[centro])
                        return centro;
            else if (num_buscado < numeros[centro])
                        return BusquedaBinaria(num_buscado,numeros,inicio,(int)((inicio+centro-1)/2),centro-1);
            else
                        return BusquedaBinaria(num_buscado,numeros,centro+1,(int)((final+centro+1)/2),final);
}
```

$$T(0) = 0$$

$$T(n) = 3 + T\left(\frac{n}{2}\right)$$

Teorema maestro:

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)\ ;\ donde\ a = 1, b = 2\ y\ f(n) = 3$$

$$O\big(f(n)\big) = O(3) = O(1)$$

$$Tomando\ el\ caso\ 2, tenemos\ que:$$

$$n^{\log_b a} = n^{\log_2 1} = n^0 = 1$$

$$Esto\ es\ asintóticamente\ igual\ a\ O\big(f(n)\big)$$

$$\therefore T(n) = \ \theta\left(n^{\log_b a}\log n\right) = \theta(\log n)$$

Algoritmo 7

```
Merge-Sort(a, p, r)
{
	if ( p < r )
	{
		q = parteEntera((p+r)/2);
		Merge-Sort(a, p, q);
		Merge-Sort(a, q+1,r);
		Merge(a, p, q, r);
	}
}
```

$$T(1) = 1$$

$$T(n) = 2T\left(\frac{n}{2}\right) + n$$

Teorema maestro:

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)\ ;\ donde\ a = 2, b = 2\ y\ f(n) = n$$

$$O\big(f(n)\big) = O(n)$$

$$Tomando\ el\ caso\ 2, tenemos\ que:$$

$$n^{\log_b a} = n^{\log_2 2} = n^1 = n$$

$$Esto\ es\ asintóticamente\ igual\ a\ O\big(f(n)\big)$$

$$\therefore T(n) = \ \theta\big(n^{\log_b a} \log n\big) = \theta(n \log n)$$

Algoritmo 8

$$T(n) = 3T\left(\frac{n}{3}\right) + 4T\left(\frac{n}{2}\right) + 2n^2 + n$$

Teorema maestro en dos partes:

$$T_1(n) = 3T\left(\frac{n}{3}\right) + n$$

$$T_1(n) = a_1 T_1\left(\frac{n}{b}\right) + f_1(n)\ ;\ donde\ a_1 = 3, b_1 = 3\ y\ f_1(n) = n$$

$$O\big(f_1(n)\big) = O(n)$$

$$Tomando\ el\ caso\ 2, tenemos\ que:$$

$$n^{\log_b a} = n^{\log_3 3} = n^1 = n$$

$$Esto\ es\ asintóticamente\ igual\ a\ O\big(f_1(n)\big)$$

$$\therefore T_1(n) = \ \theta\big(n^{\log_b a} \log n\big) = \theta(n \log n)$$

$$T_2(n) = 4T\left(\frac{n}{2}\right) + 2n^2$$

$$T_2(n) = a_2 T_2\left(\frac{n}{b}\right) + f_2(n)\ ;\ donde\ a_2 = 4, b_2 = 2\ y\ f_2(n) = 2n^2$$

$$O\big(f_2(n)\big) = O(2n^2) = O(n^2)$$

$$Tomando\ el\ caso\ 2, tenemos\ que:$$

$$n^{\log_b a} = n^{\log_2 4} = n^2$$

$$Esto\ es\ asintóticamente\ igual\ a\ O\big(f_2(n)\big)$$

$$\therefore T_2(n) = \ \theta\big(n^{\log_b a} \log n\big) = \theta(n^2 \log n)$$

Teniendo en cuenta ambos resultados, resulta que:

$$T(n) = T_1(n) + T_2(n) = \theta(n \log n) + \theta(n^2 \log n) = \theta(n^2 \log n)$$

Algoritmo 9

$$T(n) = T(n-1) + T(n-2) + T\left(\frac{n}{2}\right) \Rightarrow n > 1\,; T(0) = 5\,; T(1) = 1$$

Dividimos el problema en dos partes, primero usaremos el Teorema Maestro:

$$T_1(n) = T\left(\frac{n}{2}\right) + 0$$

$$T_1(n) = a_1 T_1\left(\frac{n}{b}\right) + f_1(n)\,;\ donde\ a_1 = 1, b_1 = 2\ y\ f_1(n) = 0$$

$$O\big(f_1(n)\big) = O(0) = O(1)$$

$$Tomando\ el\ caso\ 2, tenemos\ que:$$

$$n^{\log_b a} = n^{\log_2 1} = n^0 = 1$$

$$Esto\ es\ asintóticamente\ igual\ a\ O\big(f_1(n)\big)$$

$$\therefore T_1(n) = \ \theta\big(n^{\log_b a} \log n\big) = \theta(\log n)$$

En Segundo lugar, calcularemos la complejidad de la recurrencia homogénea:

$$T_2(n) = T(n-1) + T(n-2)$$

$$T_2(n) - T(n-1) - T(n-2) = 0$$

$$x^2 - x - 1 = 0$$

$$r_1 = \frac{1+\sqrt{5}}{2}\ ;\ r_2 = \frac{1-\sqrt{5}}{2}$$

$$T_2(n) = c_1\left(\frac{1+\sqrt{5}}{2}\right)^n + c_2\left(\frac{1-\sqrt{5}}{2}\right)^n$$

Sistemas de ecuaciones:

$$T_2(0) = c_1\left(\frac{1+\sqrt{5}}{2}\right)^0 + c_2\left(\frac{1-\sqrt{5}}{2}\right)^0 = c_1 + c_2 = 5$$

$$T_2(1) = c_1\left(\frac{1+\sqrt{5}}{2}\right)^1 + c_2\left(\frac{1-\sqrt{5}}{2}\right)^1 \approx (1.6180)c_1 + (-0.6180)c_2 = 1$$

$$c_1 \approx \frac{2045}{1118}\ ;\ c_2 \approx \frac{3545}{1118}$$

$$\therefore T_2(n) \approx \left(\frac{2045}{1118}\right)\left(\frac{1+\sqrt{5}}{2}\right)^n + \left(\frac{3545}{1118}\right)\left(\frac{1-\sqrt{5}}{2}\right)^n = O\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$$

Teniendo en cuenta ambos resultados, resulta que:

$$T(n) = T_1(n) + T_2(n) = \theta(\log n) + O\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right) = O\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$$

Algoritmo 10

$$T(n) = T\left(\frac{n}{2}\right) + 2T\left(\frac{n}{4}\right) + 2$$

Dividimos el problema en dos partes, primero usaremos el Teorema Maestro:

$$T_1(n) = T\left(\frac{n}{2}\right) + 0$$

$$T_1(n) = a_1 T_1\left(\frac{n}{b}\right) + f_1(n)\ ;\ donde\ a_1 = 1, b_1 = 2\ y\ f_1(n) = 0$$

$$O\big(f_1(n)\big) = O(0) = O(1)$$

$$Tomando\ el\ caso\ 2, tenemos\ que:$$

$$n^{\log_b a} = n^{\log_2 1} = n^0 = 1$$

$$Esto\ es\ asintóticamente\ igual\ a\ O\big(f_1(n)\big)$$

$$\therefore T_1(n) = \ \theta\big(n^{\log_b a} \log n\big) = \theta(\log n)$$

Para la segunda parte, volvemos a usar el Teorema Maestro:

$$T_2(n) = 2T\left(\frac{n}{4}\right) + 2$$

$$T_2(n) = a_2 T_2\left(\frac{n}{b}\right) + f_2(n)\ ;\ donde\ a_2 = 2, b_2 = 4\ y\ f_2(n) = 2$$

$$O\big(f_2(n)\big) = O(2) = O(1)$$

$$Tomando\ el\ caso\ 1, tenemos\ que:$$

$$n^{\log_b a - \varepsilon} = n^{\log_4 2 - 0.5} = n^0 = 1, para\ \varepsilon = 0.5$$

$$Esto\ es\ asintóticamente\ igual\ a\ O\big(f_2(n)\big)$$

$$\therefore T_2(n) = \ \theta\big(n^{\log_b a}\big) = \theta\big(\sqrt{n}\big)$$

Teniendo en cuenta ambos resultados, resulta que:

$$T(n) = T_1(n) + T_2(n) = \theta(\log n) + \theta\big(\sqrt{n}\big) = \theta\big(\sqrt{n}\big)$$