Instituto Politécnico Nacional

Escuela Superior de Cómputo

Análisis de Algoritmos

Profesor: Edgardo Adrián Franco Martínez

Alumno: Ortega Victoriano Ivan

Fecha: 24 de Marzo 2017

Análisis de algoritmos recursivos.

Ejercicio 1: Calcular la complejidad para el siguiente algoritmo.

```
#include <stdio.h>
#include <stdlib.h>

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

Sol: Del algoritmo podemos ver que si $n = 0$, entonces $T(0) = 1$, además si $n = 1$ entonces $T(1) = 5$ (comparar 1 vez el *if* de la función, 1 asignación de la variable resultado, 1 ejecución del for, 2 comparaciones del for), y por último si $n = 2$ entonces $T(2) = 9$ (se analiza igual que el caso de $n = 1$). Los análisis anteriores son para los costos de los casos base.

Ahora, analizando para un n en general, tendremos que:

$$T(n) = T(n-1) + c_{mult} + T(n-2) + c_{div} + T(n-3)$$

Donde $c_{mult} = c_{div} = 1$, luego:

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

Que se trata de una ecuación en recurrencia no homogénea.

Reacomodando la ecuación, tenemos:

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

Haciendo el cambio $T(n) = x^3$, $b = 2$ y $d = 0$, obtenemos la ecuación característica:

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

De donde las raíces son:

$$r_1 = 1.84$$
$$r_2 = -0.42 + i0.61$$
$$r_3 = -0.42 - i0.61$$
$$r_4 = 2$$

Que son raíces distintas, así:

$$T(n) = c_1(1.84)^n + c_2(-0.42 + i0.61)^n + c_3(-0.42 - i0.61)^n + c_4 2^n$$

Dado que tenemos dos números complejos, de acuerdo con sus propiedades, la ecuación puede reescribirse de la siguiente manera:

$$T(n) = c_1(1.84)^n + c_2(0.73^n(\cos(124.68) + i\sin(124.68)))$$
$$+c_3(0.73^n(\cos(124.68n) - i\sin(124.68n))) + c_4 2^n$$

Luego

$$T(n) = c_1(1.84)^n + 0.73^n((c_2+c_3)\cos(124.68n) + (c_2-c_3)i\sin(124.68n))) + c_4 2^n$$

Llamemos $k_1 = c_2 + c_3$ y $k_2 = (c_2 - c_3)i$, así

$$T(n) = c_1(1.84)^n + 0.73^n(k_1\cos(124.68n) + k_2\sin(124.68n))) + c_4 2^n$$

Evaluando las condiciones iniciales

$$T(0) = c_1 + k_1 + c_4$$
$$T(1) = 1.84c_1 - 0.41k_1 + 0.60k_2 + 2c_4$$
$$T(2) = 3.38c_1 - 0.18k_1 - 0.49k_2 + 4c_4$$

Como $T(3) = T(2) + T(1) + T(0) + 2$ entonces:

$$1 + 5 + 9 + 2 = 6.22c_1 + 0.41k_1 + 0.11k_2 + 7c_4 + 2$$

Luego

$$1 = c_1 + k_1 + c_4$$
$$5 = 1.84c_1 - 0.41k_1 + 0.60k_2 + 2c_4$$
$$9 = 3.38c_1 - 0.18k_1 - 0.49k_2 + 4c_4$$
$$15 = 6.22c_1 + 0.41k_1 + 0.11k_2 + 7c_4$$

Que es el sistema de ecuaciones resultante de la evaluacion de las condiciones iniciales.
Pasando este sistema a la matriz aumentada, tenemos:

$$\left(\begin{array}{cccc|c} 1 & 1 & 0 & 1 & 1 \\ 1.84 & -0.41 & 0.60 & 2 & 5 \\ 3.38 & -0.18 & -0.49 & 4 & 9 \\ 6.22 & 0.41 & 0.11 & 7 & 15 \end{array}\right) \tag{1}$$

Resolviendo la matriz por Gauss-Jordan, se llega a la solución:

$$\left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & -1.581 \\ 0 & 1 & 0 & 0 & -1.018 \\ 0 & 0 & 1 & 0 & 0.486 \\ 0 & 0 & 0 & 1 & 3.6 \end{array}\right) \tag{2}$$

Que es equivalente a que

$$c_1 = -1.581$$

$$k_1 = -1.018$$

$$k_2 = 0.486$$

$$c_4 = 3.6$$

$\therefore T(n) = -1.581(1.84)^n + 0.73^n[(0.486)\sin(n\theta) - (1.018)\cos(n\theta)] + (3.6)2^n$
En donde $\theta = 124.68$

Comentario: Al parecer hay unos cuantos errores en cuanto a lo que hace el algoritmo, de hecho, en la parte donde $n < 3$, al hacer a $resultado = 0$, este valor nunca va a cambiar, ya que en el loop *for* se hace la operación $resultado* = num$, pero como $resultado = 0$, siempre va a retornar un valor de cero. De lo que a partir de un $n > 3$ va a haber divisiones entre 0, lo cual será un error en el programa.

Ejercicio 2: Calcular la complejidad de la implementación recursiva del termino n de la serie de Tribonacci (0, 1, 1, 2, 4, 7, 13, 24, 44, 81, 149, 274, 504, 927, 1705, ...)

```
#include <stdio.h>
#include <stdlib.h>

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);
}
```

Sol: Si nos damos cuenta, el análisis es similar al caso anterior, solo hay ligeras modificaciones en cuanto a las conidiciones iniciales y a la ecuacion en recurrencia.
Es evidente que $T(0) = 1$, $T(1) = T(2) = 2$, ya que son las comparaciones que se hacen al entrar a la función.
Ahora para un n en general:

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

Donde el 4 resulta de las dos comparaciones de la función, mas el costo de las 2 sumas.
Que avanzando un poco más (ya que el análisis es exactamente igual al del ejercicio anterior), la ecuación característica quedaría de la siguiente manera:

$$(x^3 - x^2 - x - 1)(x - 4) = 0$$

Donde nuestras raíces ahora son:

$$r_1 = 1.84$$

$$r_2 = -0.42 + i0.61$$

$$r_3 = -0.42 - i0.61$$

$$r_4 = 4$$

Siguiendo el análisis del algoritmo anterior, llegaremos a que la complejidad de la función, estará dada por:

$$T(n) = c_1(1.84)^n + 0.73^n(k_1\cos(124.68n) + k_2\sin(124.68n))) + c_4 4^n$$

Resolviendo las condiciones iniciales:

$$T(0) = c_1 + k_1 + c_4$$

$$T(1) = 1.84c_1 - 0.41k_1 + 0.60k_2 + 4c_4$$

$$T(2) = 3.38c_1 - 0.18k_1 - 0.49k_2 + 8c_4$$

Como $T(3) = T(2) + T(1) + T(0) + 4$ entonces:

$$1 + 2 + 2 + 4 = 6.22c_1 + 0.41k_1 + 0.11k_2 + 13c_4 + 4$$

Luego

$$1 = c_1 + k_1 + c_4$$

$$2 = 1.84c_1 - 0.41k_1 + 0.60k_2 + 4c_4$$

$$2 = 3.38c_1 - 0.18k_1 - 0.49k_2 + 8c_4$$

$$5 = 6.22c_1 + 0.41k_1 + 0.11k_2 + 13c_4$$

Pasando este sistema a la matriz aumentada, tenemos:

$$\left(\begin{array}{cccc|c} 1 & 1 & 0 & 1 & 1 \\ 1.84 & -0.41 & 0.60 & 4 & 2 \\ 3.38 & -0.18 & -0.49 & 8 & 2 \\ 6.22 & 0.41 & 0.11 & 13 & 5 \end{array}\right) \tag{3}$$

Resolviendo la matriz por Gauss-Jordan, se llega a la solución:

$$\left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 1.679 \\ 0 & 1 & 0 & 0 & 0.120 \\ 0 & 0 & 1 & 0 & 0.930 \\ 0 & 0 & 0 & 1 & -0.8 \end{array}\right) \tag{4}$$

Que es equivalente a que

$$c_1 = 1.679$$

$$k_1 = 0.120$$

$$k_2 = 0.930$$

$$c_4 = -0.8$$

$$\therefore T(n) = 1.679(1.84)^n + 0.73^n[(0.120)\cos(n\theta) + (0.930)\sin(n\theta)] - (0.8)2^n$$
$$\text{Con } \theta = 124.68$$

Ejercicio 3: Resolver las siguientes ecuaciones y dar su orden de complejidad:

$$T(n) = 3T(n-1) + 4T(n-2) \text{ si } n > 1;\ T(0) = 0,\ T(1) = 1$$

$$T(n) = 2T(n-1) - (n+5)3^n \text{ si } n > 0;\ T(0) = 0$$

$$T(n) = 3T(n-1) + 4T(n-2) + (n+5)2^n \text{ si } n > 1;\ T(0) = 0,\ T(1) = 100$$

$$T(n) - 2T(n-1) = 3^n \text{ si } n \geq 2;\ T(0) = 0,\ T(1) = 1$$

Ejercicio 4: Calcular la cota de complejidad del algoritmo de búsqueda binaria recursiva:

Ejercicio 5: Calcular la cota de complejidad del algoritmo Merge-Sort recursivo: