INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

ANALISIS DE ALGORITMOS

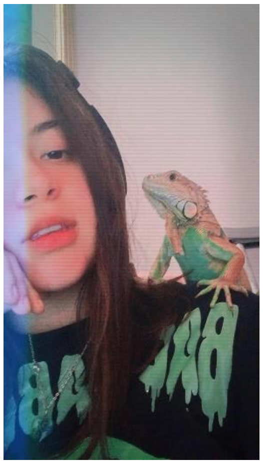
PROFESOR TITULAR: FRANCO MARTINEZ EDGARDO ADRIAN

1° PARCIAL

EJERCICIOS #9:
ANALISIS DE ALGORITMOS RECURSIVOS

LEMUS RUIZ MARIANA ELIZABETH
2020630211

GRUPO: 3CM12



LEMUS RUIZ MARIANA ELIZABETH
3CM12

EJERCICIOS 09: ANALISIS DE ALGORITMOS RECURSIVOS

INSTRUCCIONES:
Para los siguientes 6 algoritmos determine la cota de complejidad O (cota superior ajustada) bajo el principio del peor caso, de cada algoritmo. Indique a detalle el procedimiento de obtención de la cota.

EJERCICIO 01

1

```
int coef(int n,int k)
{
    if (k==0||k==n)          -> 2 } 3
        return 1;            -> 1
    else if (k>0&&k<n)       -> 2(3) } 7
        return coef(n-1,k-1)+coef(n-1,k);   1
}
```

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

Para $\begin{cases} n=0 \longrightarrow T(n) = 3 \\ n>0 \longrightarrow T(n) = 7 + 2T(n-1) \end{cases}$

Sustituimos $T(n) = x \longrightarrow T(n) - 2T(n-1) = 7$

$$x - 2 = 7$$

$$(x-2)(x-1) = 0 \quad \begin{cases} r_1 = 2 \\ r_2 = 1 \end{cases}$$

Tomamos:

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

De condiciones iniciales:

$$T(0) = c_1 + c_2 = 3 \longrightarrow c_2 = 3 - c_1$$

$$T(1) = 7 + 2(3) = 13 \longrightarrow 2c_1 + 3 - c_1 = 13$$

$$c_1 = 13 - 3 = 10$$

$$c_2 = 3 - 10 = -7$$

Sustituyendo

$$T(n) = 10(2^n) + (-7)1^n \in O(c^n)$$

LEMUS RUIZ MARIANA ELIZABETH
3CM12

EJERCICIO 03

(3)

```
Palindromo(cadena)
{
    if(longitud(cadena)==1)                                  → 2 } 3
        return TRUE                                          → 1
    if(primer_caracter(cadena)!=ultimo_caracter(cadena))     → 2 }
        return FALSE                                         → 1  } 4
    cadena=remover_primer_ultimo_caracter(cadena)            → 2  }
    Palindromo(cadena)                                       → T(n-2)
}
```

$$\text{Si}\begin{cases} n=1 & T(n)=3 \\ n>1 & T(n)=4+T(n-2) \in O(n) \end{cases}$$

EJERCICIO 04

(4)

```
SubAlgoritmo Volados(n,cadena)
    Si n!=0                                    → 1
        Volados(n-1,concatenar(cadena,'S'))    → T(n-1) }
        Volados(n-1,concatenar(cadena,'A'))    → T(n-1) } 2T(n-1)
    SiNo
        Mostrar cadena                         → 1
    FinSi
FinSubAlgoritmo
```

$$\text{if}\begin{cases} 0=n \rightarrow T(n)=1 \\ 0>n \rightarrow T(n)=1+2T(n-1) \end{cases} \qquad \text{Sustituimos } T(n)=x$$

Entones $T(n)-2T(n-1)=1 \rightarrow x-2=1$ $\quad r_1=2$

$(x-2)(x-1)=0$ $\quad r_2=1$

Tomando

$T(n)=C_1(2^n)+C_2(1^n)$

De condiciones

$T(1)=2C_1+C_2=1 \longrightarrow C_2=1-2C_1$

$T(2)=4C_1+C_2=3 \longrightarrow 4C_1-2C_1+1=3$

$2C_1=2$

$C_1=1, \quad C_2=-1$

Sustiuimos:

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

$=2^n-1^n \in \underline{\underline{O(c^n)}}$

EJERCICIO 05

5

```
DecABin(n)
{
    if(n>1)          → 1
    DecABin(n/2)     → T(n/2)
    Mostrar(n%2)
}
```

Para $\begin{cases} n=0 & T(n)=1 \\ n>0 & T(n)=1+T(n/2) \end{cases}$

Usando el teorema maestro con $\begin{cases} a=1 \\ b=1/2 \\ f(n)=1 \end{cases}$

Entonces:

$n^{\log_b a} = n^{\log_{1/2} 1} = n^0 = 1$

$\therefore T(n) = \Theta(\log n)$

EJERCICIO 06

6

```
int Producto( int a, int b)
{
    if (b==0)        → 1
        return 0;    → 1
    else
        return a + Producto(a,b-1);
                 ↓         ↓
                 1       T(n-1)
}
```

if $\begin{cases} n=0 \rightarrow T(n)=2 \\ n>0 \rightarrow T(n)=2+T(n-1) \end{cases}$

$\therefore \in O(n)$