INSTITUTO POLITECNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO (ESCOM)

ANÁLISIS DE ALGORITMOS

NOMBRE DEL ALUMNO:

- SANTOS MÉNDEZ ULISES JESÚS

EJERCICIO 09:

- ANÁLISIS DE ALGORITMOS RECURSIVOS

FECHA DE ENTREGA:

- 15/04/2022

GRUPO:

- 3CM14

Análisis de algoritmos recursivos

Objetivo:

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.

Código 01:

```
Busqueda(A[],i,val)
{
    if(i<0)
        return -1
    if(A[i]==val)
        return i
    return Busqueda(A[],i-1,val)
}
```

$T(0) = 3$

$T(n) - T(n-1) = 4 \quad d = 0 \;\; b = 1 \quad k = 1$

$(x-1)^2 = 0$

$r_1 = 1 \quad r_2 = 1$

$$T(n) = \sum_{i=1}^{m} C_k \, n^{(i-1)} r^n$$

$T(n) = \; C_1 n^{1-1} 1^n + C_2 n^{2-1} 1^n$

$T(n) = \; C_1 + C_2 n$

$T(0) = C_1$

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

$T(2) = 4 + T(1) = 4 + 7 = 11$

$11 = 3 + 2C_2$

$C_2 = 4$

$T(n) = 3 + 4n \quad \in O(n)$

Código 02:

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

$T(0) = 1$

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

$T(n) - 2T(n-1) = 4 \qquad b = 1 \quad d = 0 \quad k = 1$

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

$r_1 = 2 \quad r_2 = 1$

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

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

$T(0) = C_1 + C_2$

$T(1) = 2C_1 + C_2$

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

$T(1) = 6$

$C_1 + C_2 = 1$

$2C_1 + C_2 = 6$

$T(n) = 5 * 2^n - 4 \qquad \in \quad O(2^n)$

Código 03:

```
Palindromo(cadena)
{
    if(longitud(cadena)==1)
        return TRUE
    if(primer_caracter(cadena)!=ultimo_caracter(cadena))
        return FALSE

    cadena=remover_primer_ultimo_caracter(cadena)
    Palindromo(cadena)
}
```

$T(0) = 3 \qquad T(1) = 4$

$T(n) - T(n-1) = 5 \quad d = 0 \;\; b = 1 \qquad k = 1$

$(x-1)^2 = 0$

$r_1 = 1 \qquad r_2 = 1$

$$T(n) = \sum_{i=1}^{m} C_k\, n^{(i-1)} r^n$$

$T(n) = \; C_1 n^{1-1} 1^n + C_2 n^{2-1} 1^n$

$T(n) = \; C_1 + C_2 n$

$T(0) = C_1$

$T(1) = C_1 + C_2$

$C_2 = 4 - C_1$

$C_2 = 1$

$T(n) = 3 + n \qquad \in O(n)$

Código 04:

```
SubAlgoritmo Volados(n,cadena)
    Si n!=0
        Volados(n-1,concatenar(cadena,'S'))
        Volados(n-1,concatenar(cadena,'A'))
    SiNo
        Mostrar cadena
    FinSi
FinSubAlgoritmo
```

$T(0) = 4 \quad T(1) = 3$

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

$T(n) - 2T(n-1) = 3 \qquad b = 1 \quad d = 0 \quad k = 1$

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

$r_1 = 2 \quad r_2 = 1$

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

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

$T(0) = C_1 + C_2$

$T(1) = 2C_1 + C_2$

$C_1 + C_2 = 4$

$2C_1 + C_2 = 3$

$C_1 = -1 \quad C_2 = 5$

$T(n) = -2^n + 5 \quad \in \ O(2^n)$

Código 05:

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

$T(0) = 1$

$T(1) = 1$

$T(2) = 2$

$T(n) = 3 + T\left(\frac{n}{2}\right) \qquad a = 1 \quad b = 2 \qquad f(n) = 3 \quad \rightarrow \quad \theta(1) = \theta(n^0)$

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

Por medio del teorema maestro tenemos:

$T(n) = \theta(n^0 \lg(n)) = \theta(\lg(n))$

Código 06:

```
int Producto( int a, int b)
{

    if (b==0)
        return 0;
    else
        return a + Producto(a,b-1);
}
```

$T(0) = 1$

$T(1) = 2$

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

$T(n) - T(n-1) = 3 \qquad b = 1 \quad d = 0 \quad k = 1$

$(x-1)^2 = 0$

$r_1 = 1 \quad r_2 = 1$

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

$T(n) = C_1 + C_2 n$

$T(0) = C_1 + C_2(0) = C_1$

$T(1) = C_1 + C_2$

$C_2 = 2 - 1 = 1$

$T(n) = 1 + n \qquad \in O(n)$