ANÁLISIS DE ALGORITMOS

Alumno: Martín Eduardo Barriga Vargas

Grupo 3CM3

Ejercicio 04: Análisis de algoritmos no recursivos

29 de marzo de 2019

A. Para los siguientes 15 algoritmos determine la cota O ( ).

1

```
for(i=1;i<n;i++)
        for(j=0;j<n-1;j++)
        {
                temp = A[j];
                A[j] = A[j+1];
                A[j+1] = temp;
        }
```

O(1) O(N) O(N^2)

2

```
polinomio=0;
for(i=0;i<=n;i++)
{
        polinomio=polinomio*z + A[n-i];
}
```

O(1) O(N)

3

```
for i = 1 to n do
        for j = 1 to n do
                C[i,j] = 0;
                for k = 1 to n do
                        C[i,j] = C[i,j] + A[i,k]*B[k,j];
```

O(N^3) O(N^2) O(N)

4

```
anterior = 1;
actual = 1;
while (n>2)
{

    aux = anterior + actual;
    anterior = actual;
    actual = aux;
    n = n - 1;
}
```

O(1)

O(1)

O(N)

5

```
for (i = n - 1,j=0;i>=0;i--,j++)
            s2[j]= s[i];
```

O(1)

O(N^2)

```
for (i = 0,i<n;i++)
            s[i]= s2[i];
```

O(1)

O(N^2)

9

```
func Producto2Mayores(A,n)
if(A[1] > A[2])
            mayor1 = A[1];
            mayor2 = A[2];
else
            mayor1 = A[2];
            mayor2 = A[1];
i = 3;

while(i<=n)
            if(A[i] > mayor1)
                        mayor2 = mayor1;
                        mayor1 = A[i];
            else if (A[i] > mayor2)
                        mayor2 = A[i];
            i = i + 1;

return = mayor1 * mayor2;
fin
```

O(1)

O(1)

O(N^2)

O(1)

10

```
func OrdenamientoIntercambio(a,n)
for (i=0; i<n-1; i++)
            for (int j=i+1; j<n;j++)
                        if (a[ j ] < a[ i ])
                        {
                                    temp=a[ i ];
                                    a[ i ]=a[ j ];
                                    a[ j ]=temp;
                        }
fin
```

O(N^2)

O(1)

O(N)

11

```
func MaximoComunDivisor(m, n)
{
        a=max(n,m);
        b=min(n,m);          } O(1)
        residuo=1;
        mientras (residuo > 0)
        {
O(log n)        residuo=a mod b;
                a=b;                  } O(1)
                b=residuo;
        }
        MaximoComunDivisor=a;
        return MaximoComunDivisor;   } O(1)
}
```

12

```
Procedimiento BurbujaOptimizada(A,n)
    O(1)  { cambios = "No"
            i=0
            Mientras i< n-1 && cambios != "No" hacer
O(N^2)              cambios = "No"
                    Para j=0 hasta (n-2)-i hacer
O(N)                        Si(A[i] < A[j]) hacer
O(1)                                aux = A[j]
                                    A[j] = A[i]
                                    A[i] = aux
                                    cambios = "Si"
                            FinSi
                    FinPara
                     i= i+1
            FinMientras
fin Procedimiento
```

13

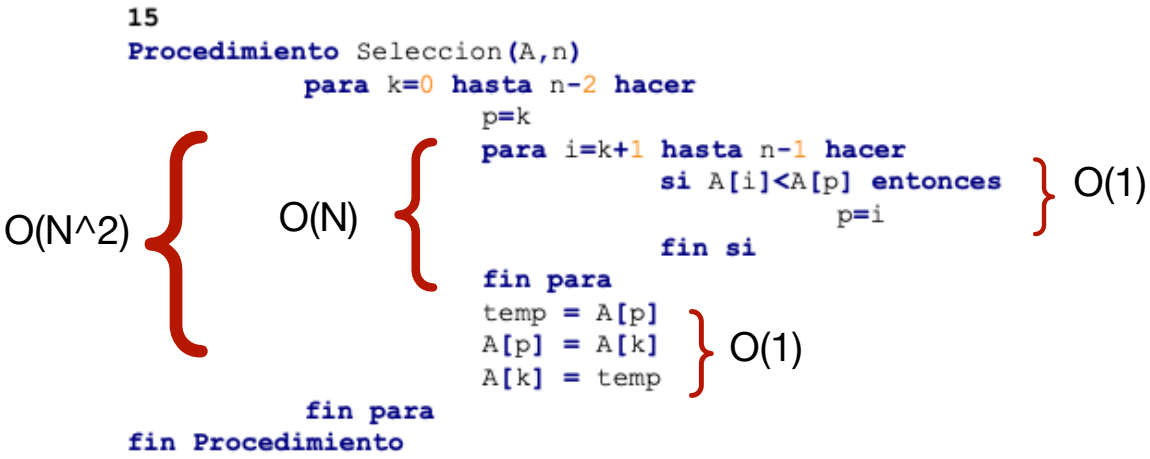
```
Procedimiento BurbujaSimple(A,n)
        para i=0 hasta n-2 hacer
                para j=0 hasta (n-2)-i hacer
                        si (A[j]>A[j+1]) entonces
                                aux = A[j]
                                A[j] = A[j+1]
                                A[j+1] = aux
                        fin si
                fin para
        fin para
fin Procedimiento
```

O(N^2) = O(N) · O(1)

14

```
Procedimiento Ordena(a,b,c)
{
        if(a>b)
                if(a>c)
                        if(b>c)
                                salida(a,b,c);
                        else
                                salida(a,c,b);
                else
                        salida(c,a,b);
        else
                if(b>c)
                        if(a>c)
                                salida(b,a,c);
                        else
                                salida(b,c,a);
                else
                        salida(c,b,a);
}
```

O(1) = O(1) · O(1) + O(1) · O(1)



Conclusión

Los algoritmos que se analizaron en este ejercicio son los mismos que en el ejercicio 1, y su análisis es más sencillo, pues sólo se está determinando cuál es el valor de la cota O(), en cambio en el ejercicio pasado teníamos que contar a detalle el número de instrucciones que se llevaban a cabo. Los resultados entre los ejercicios varían, sin embargo se puede llegar al mismo orden aproximando una función polinomial con los resultados del ejercicio pasado y multiplicando esto por alguna constante.