Dibujo en blanco y negro

Descripción generada automáticamente con confianza baja**INSTITUTO POLITÉCNICO NACIONAL**

**ESCUELA SUPERIOR DE CÓMPUTO**

**--------------------ANÁLISIS DE ALGORITMOS------------------**

**ACTIVIDAD**

Análisis de algoritmos recursivos

**PROFESOR:**

Franco Martínez Edgardo Adrián

**ALUMNO:**

Meza Vargas Brandon David – 2020630288

**GRUPO:**

3CM13

Joven con camiseta negra

Descripción generada automáticamente

**Índice**

[Código 01 3](#_Toc84845335)

[Código 02 5](#_Toc84845336)

[Código 03 7](#_Toc84845337)

[Código 04 9](#_Toc84845338)

[Código 05 11](#_Toc84845339)

[Código 06 12](#_Toc84845340)

Para los siguientes algoritmos considerarán las siguientes operaciones:

* Asignaciones y returns
* Aritméticas
* Comparación

# **Código 01**

Texto

Descripción generada automáticamente

En la siguiente imagen podemos ver la cantidad de operaciones que tenemos por cada línea:

Texto

Descripción generada automáticamente

Podemos ver que tenemos un primer caso en donde i es menor que 0, en este caso tenemos:

T(0)=2, la comparación y el return

En caso contrario tenemos

T(n) = 1+2´T(n-1)

De lo anterior tenemos la comparación del primer if, la comparación y return del segundo if cuando el valor es encontrado y se le suma la llamada a la misma función con una transformación, la cual es que se llama al mismo problema reducido en uno, de esta forma tenemos:

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

Con el modelo recurrente anterior podemos hallar su equivalente evaluando para alguna n, en este ejemplo lo haremos para n=5 sabiendo que T(0)=2:

T(0)=2

T(1)=3+T(0)=3+2=5

T(2)=3+T(1)=3+5=8

T(3)=3+T(2)=3+8=11

T(4)=3+T(3)=3+11=14

T(5)=3+T(4)=17

Analizando el comportamiento anterior nos damos cuenta de que podemos hallar un modelo sin recurrencia y su cota:

***T(n)=3n+2* ∈ O(n)**

# **Código 02**

Texto

Descripción generada automáticamente

En la siguiente imagen podemos ver la cantidad de operaciones que tenemos por cada línea:

Texto

Descripción generada automáticamente con confianza media

Podemos ver que el caso base es donde k es igual a 0 o igual a n, en este caso tenemos lo siguiente:

T(0)=2 comparación y return

En caso contrario tenemos lo siguiente:

T(n)=1+2+5+T(n-1)+T(n-1) = 8+2T(n-1)

Lo anterior se obtuvo de la primera comparación, posteriormente las dos comparaciones que se hacen en el segundo if y las 5 operaciones que tenemos al final, además sumamos dos veces la llamada a la misma función pero reducida en 1.

Para calcular la cota de este algoritmo obtendremos un modelo no recurrente equivalente:

Sustituyendo

T(n) = x b = 1 d = 0

Así tenemos:

Obteniendo las raíces

r1 = 1 r2 = 2

Si T(0) = 2 y T(1) = 8+4 = 12

De esta forma calculando los coeficientes:

c1 = -8 c2 = 10

Por lo tanto:

# **Código 03**

Texto

Descripción generada automáticamente

En la siguiente imagen podemos ver la cantidad de operaciones que tenemos por cada línea:

Texto

Descripción generada automáticamente

Tenemos el caso base donde la longitud de la cadena es igual a 1, en esta parte tenemos:

T(0)=2 la comparación y el return

Para los demás casos tenemos lo siguiente:

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

Aquí contamos la primera comparación, la segunda comparación, la asignación a la cadena y la llamada transformada a la misma función con n-1, este -1 es por que se llama a la misma función haciendo una transformación a la cadena.

Aquí para calcular la cota pasamos a una función no recurrente de la siguiente forma:

T(0)=2

T(1)=3+T(0)=3+2=5

T(2)=3+T(1)=3+5=8

T(3)=3+T(2)=3+8=11

T(4)=3+T(3)=3+11=14

T(5)=3+T(4)=17

Analizando el comportamiento anterior nos damos cuenta de que podemos hallar un modelo sin recurrencia y su cota:

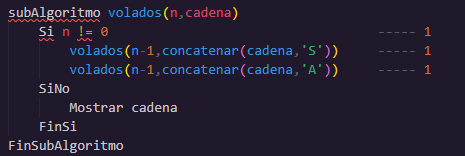
***T(n)=3n+2* ∈ O(n)**

# **Código 04**

Texto

Descripción generada automáticamente

En la siguiente imagen podemos ver la cantidad de operaciones que tenemos por cada línea:



Tenemos un caso base el cual es:

T(0)=1, solo es la comparación del inicio.

Para lo siguiente tenemos que es:

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

En lo anterior contamos la comparación y las dos operaciones que están en la recursión, de igual forma contamos la transformación.

Para calcular la cota de este algoritmo obtendremos un modelo no recurrente equivalente:

Sustituyendo

T(n) = x b = 1 d = 0

Así tenemos:

Obteniendo las raíces

r1 = 1 r2 = 2

Si T(0) = 1 y T(1) = 3+2 = 5

De esta forma calculando los coeficientes:

c1 = -3 c2 = 4

Por lo tanto:

# **Código 05**

Texto

Descripción generada automáticamente

En la siguiente imagen podemos ver la cantidad de operaciones que tenemos por cada línea:

Texto

Descripción generada automáticamente

Para el caso base donde n no sea mayor a 1, tendremos:

T(1)=1 donde solo se cuenta la comparación

Para los demás casos tendremos:

T(n)=1+2+T(n/2)=3+T(n/2)

De lo anterior se cuenta la comparación y las dos operaciones que se tienen dentro del if, además de la transformación que se hace.

En este algoritmo hacemos uso del **teorema maestro.**

De esta manera tenemos que

a=1 b=2 f(n)=3

Por lo tanto aplicando el límite asintótico tenemos:

De esta manera aplicamos el segundo caso del teorema maestro ya que:

Así tenemos finalmente:

# **Código 06**

Texto

Descripción generada automáticamente

En la siguiente imagen podemos ver la cantidad de operaciones que tenemos por cada línea:

Texto

Descripción generada automáticamente

Aquí tenemos el caso cuando b es igual a 0, en este caso tenemos:

T(0)=2, la comparación y el return

En caso contrario:

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

De lo anterior se cuenta la primera comparación y las 3 operaciones dentro del else.

Así podemos encontrar una función no recurrente equivalente y de igual forma la cota haciendo algunas evaluaciones:

T(5)=4+T(4)=4+18=22

T(4)=4+T(3)=4+14=18

T(3)=4+T(2)=4+10=14

T(2)=4+T(1)=4+6=10

T(1)=4+T(0)=4+2=6

Analizando lo anterior llegamos a:

***T(n)=4n+2 O(n)***