INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

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

ACTIVIDAD

Número de impresiones

PROFESOR:

Franco Martínez Edgardo Adrián

ALUMNO:

Meza Vargas Brandon David – 2020630288

GRUPO:

3CM13

Código 01

```c
    int i,j=0,n;

    if(argc != 2){
        printf("\nIntroduce el valor de n");
        exit(1);
    }

    n = atoi(argv[1]);

    for(i=10; i<n*5; i*=2)
        printf("\n%dAlgoritmos", ++j);
}
```

Función

El comportamiento del ciclo es exponencial cuando se dice que i incrementa de la siguiente forma: **i*=2,** por lo tanto, tenemos un comportamiento exponencial, pero también debemos tener en cuenta el límite que es 5n, de esta forma nos queda la siguiente función piso:

$$f(n) = \lfloor \log_2 5n \rfloor$$

Pero no queda ahí, sino que también debemos tener en cuenta los saltos que dio al inicio, de esta forma restárselo a la función anterior para así obtener el número de impresiones correctos, siendo los primeros saltos:

$$f(n) = \lfloor \log_2 10 \rfloor$$

Así tenemos que:

$$\boldsymbol{f(n) = \lfloor \log_2 5n \rfloor - \lfloor \log_2 10 \rfloor}$$

Tabla Comparativa

N	Resultados Teóricos	Resultados Empíricos
-1	**Math error**	**0**
0	**Math error**	**0**
1	**0**	**0**
2	**0**	**0**
3	**1**	**1**
5	**2**	**2**
15	**3**	**3**
20	**4**	**4**
100	**6**	**6**

409	**8**	**8**
500	**8**	**8**
593	**9**	**9**
1000	**9**	**9**
1471	**10**	**10**
1500	**10**	**10**
2801	**11**	**11**
3000	**11**	**11**
5000	**12**	**12**
10000	**13**	**13**
20000	**14**	**14**

Gráfica

Código 02

```c
int main (int argc, char ** argv){
    int i,j=0,n, k=0;

    if(argc != 2){
        printf("\nIntroduce el valor de n");
        exit(1);
    }

    n = atoi(argv[1]);

    for(j=n; j>1; j/=2){
        if(j<(n/2)){
            for(i=0; i<n; i+=2)
                printf("\n%dAlgoritmos", ++k);
        }
    }
```

Función

Podemos observar que el for que está más adentro tiene un comportamiento lineal, resultando en la función techo:

$$f(n) = \lceil n/2 \rceil$$

Posteriormente, vemos que el primer for va a ir reduciendo en j/=2 y dentro comienza con un if que nos da otra condición de que sea menor a la mitad de n, junto con esta y la propia condición del for tenemos 2, las cuales representan dos valores menos al valor resultante de la función de este for, siendo entonces la función piso:

$$f(n) = \lfloor \log_2(n) - 2 \rfloor$$

De esta forma podemos determinar la función final del código:

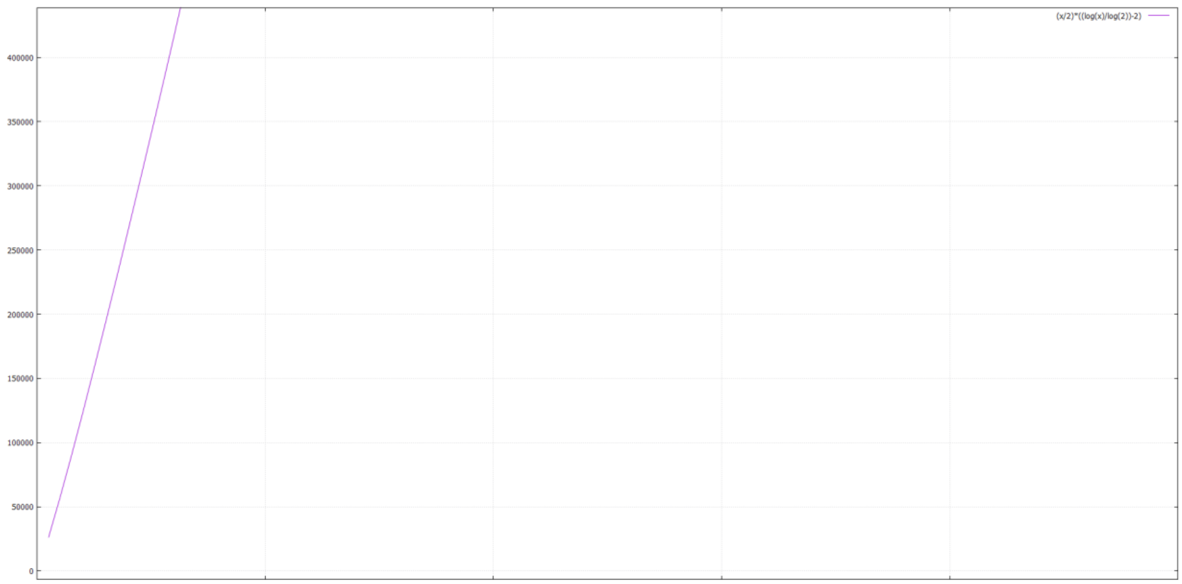
$$f(n) = \left\lceil \frac{n}{2} \right\rceil \lfloor (\log_2(n) - 2) \rfloor$$

Tabla Comparativa

N	Resultados Teóricos	Resultados Empíricos
-1	**0**	**0**
0	**0**	**0**
1	**0**	**0**
2	**-1**	**0**
3	**0**	**0**
5	**0**	**0**
15	**8**	**8**

20	**20**	**20**
100	**200**	**200**
409	**123**	**123**
500	**1500**	**1500**
593	**2079**	**2079**
1000	**3500**	**3500**
1471	**5888**	**5888**
1500	**6000**	**6000**
2801	**12609**	**12609**
3000	**13500**	**13500**
5000	**25000**	**25000**
10000	**55000**	**55000**
20000	**120000**	**120000**

Gráfica



Código 03

```c
int main (int argc, char ** argv){
    int i,j=0,n, k=0, l=0;

    if(argc != 2){
        printf("\nIntroduce el valor de n");
        exit(1);
    }

    n = atoi(argv[1]);

    for(i=0; i<n*5; i+=2){
        for(j=0; j<2*n; j++){
            for(k=j; k<n; k++)
                printf("\n%dAlgoritmos", ++l);
        }
    }
}
```

Función

Primeramente, comenzamos analizando los dos for internos del código, si lo analizamos bien es una serie:

$$\sum_{j=0}^{n} n-j$$

Cuya convergencia es:

$$\sum_{j=0}^{n} n-j = \frac{n^2-n}{2}+n = \left\lfloor\frac{n^2+n}{2}\right\rfloor$$

Analizando el primer for, tenemos que su función correspondiente es la función techo:

$$f(n) = \left\lceil\frac{5n}{2}\right\rceil$$

De esta forma, multiplicando, podemos concluir que la función del código es:

$$f(n) = \left\lceil\frac{5n}{2}\right\rceil \left\lfloor\frac{n^2+n}{2}\right\rfloor$$

Tabla Comparativa

N	Resultados Teóricos	Resultados Empíricos
-1	**0**	**0**
0	**0**	**0**
1	**3**	**3**
2	**15**	**15**
3	**48**	**48**
5	**195**	**195**
15	**4560**	**4560**
20	**10500**	**10500**
100	**1262500**	**1262500**
409	**85773435**	**85773435**
500	**156562500**	**156562500**
593	**261187443**	**261187443**
1000	**1251250000**	**1251250000**
1471	**3982008768**	**3982008768**
1500	**4221562500**	**4221562500**
2801	$\mathbf{2.74811796x10^{10}}$	$\mathbf{2.74811796x10^{10}}$
3000	$\mathbf{3.376125x10^{10}}$	$\mathbf{3.376125x10^{10}}$
5000	$\mathbf{1.5628125x10^{11}}$	$\mathbf{1.5628125x10^{11}}$
10000	$\mathbf{1.250125x10^{12}}$	$\mathbf{1.250125x10^{12}}$
20000	$\mathbf{1.00005x10^{13}}$	$\mathbf{1.00005x10^{13}}$

En los resultados Empíricos, a partir de la N = 1000, se suponen correctos los resultados ya que los anteriores fueron todos correctos, sin embargo, a partir de 1000 el proceso es muy tardado debido a la cantidad de impresiones que realiza.

Gráfica

Código 04

```c
int main (int argc, char ** argv){
    int i,j=0,n, k=0;

    if(argc != 2){
        printf("\nIntroduce el valor de n");
        exit(1);
    }

    n = atoi(argv[1]);

    i = n;
    while(i >= 0){
        for(j=n; i<j; i-=2, j/=2)
            printf("\n%dAlgoritmos", ++k);
    }
}
```

Función

Si analizamos bien el código nos damos cuenta de que no se imprimirá en ningún momento la palabra algoritmos puesto que antes de entrar al bucle while igualamos i a n, esto provoca que la condición del for dentro de while nunca se cumpla, pues debido a que j empieza en n, i y j siempre serán iguales.

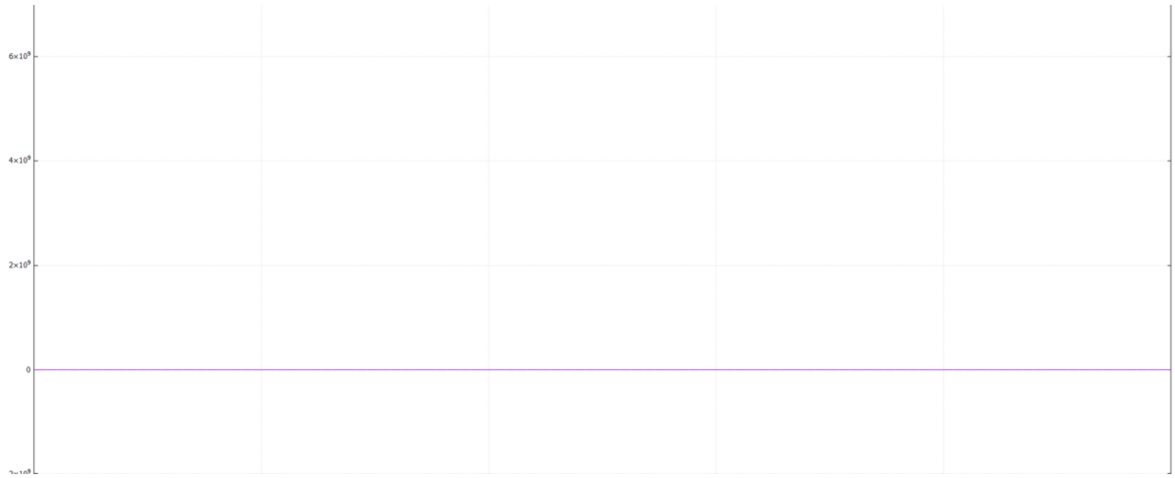
$$f(n) = 0$$

Tabla Comparativa

N	Resultados Teóricos	Resultados Empíricos
-1	**0**	**0**
0	**0**	**0**
1	**0**	**0**
2	**0**	**0**
3	**0**	**0**
5	**0**	**0**
15	**0**	**0**
20	**0**	**0**
100	**0**	**0**
409	**0**	**0**
500	**0**	**0**
593	**0**	**0**

1000	**0**	**0**
1471	**0**	**0**
1500	**0**	**0**
2801	**0**	**0**
3000	**0**	**0**
5000	**0**	**0**
10000	**0**	**0**
20000	**0**	**0**

Gráfica



Código 05

```c
int main (int argc, char ** argv){
    int i,j=0,n, k=0;

    if(argc != 2){
        printf("\nIntroduce el valor de n");
        exit(1);
    }

    n = atoi(argv[1]);

    for(i=1;i<4*n; i*=2)
        for(j=i; j<5*n; j+=3)
            printf("\n%dAlgoritmos", ++k);

}
```

Función

Primeramente, si analizamos el for interno vemos que j depende de i, esto nos lleva a pensar en una sumatoria, ya que el modelo del for interno nos quedaría de la siguiente forma:

$$\boldsymbol{f(n) = \left\lceil \frac{5n - i}{3} \right\rceil}$$

Es importante recordar que la función será techo, pero tenemos un detalle, la i avanza de forma exponencial como lo indica el for más externo, es decir, va aumentando en potencias de 2:

$$\boldsymbol{\left\lceil \frac{5n-0}{3} \right\rceil + \left\lceil \frac{5n-2}{3} \right\rceil + \left\lceil \frac{5n-4}{3} \right\rceil + \left\lceil \frac{5n-8}{3} \right\rceil + \left\lceil \frac{5n-16}{3} \right\rceil \ldots}$$

Por lo tanto, nos lleva a pensar en una sumatoria, pero viendo el primer for vemos que la i no va a llegar hasta n, ya que como se menciona, su comportamiento es exponencial, por lo tanto, el límite superior de la sumatoria seria:

$$\boldsymbol{f(n) = \lfloor \log_2(4n) \rfloor --- \textit{Siendo función piso}}$$

Siendo la función resultante del código la siguiente, tengamos en cuenta que debemos de sumar el caso donde j inicia en 1:

$$\boldsymbol{f(n) = \left\lfloor \frac{5n}{3} \right\rfloor + \sum_{i=1}^{\lfloor \log_2(4n) \rfloor} \left\lceil \frac{5n - 2^i}{3} \right\rceil}$$

Lo comprobaremos “a mano” para N= 5, 15 y 20:

N=5

$$\boldsymbol{f(5) = \left\lfloor \frac{5(5)}{3} \right\rfloor + \sum_{i=1}^{4} \left\lceil \frac{5n - 2^i}{3} \right\rceil = \left\lceil \frac{5(5) - 2^1}{3} \right\rceil + \left\lceil \frac{5(5) - 2^2}{3} \right\rceil + \left\lceil \frac{5(5) - 2^3}{3} \right\rceil + \left\lceil \frac{5(5) - 2^4}{3} \right\rceil}$$
$$\boldsymbol{= \lfloor 8.33 \rfloor + 8 + 7 + 6 + 3 = 32}$$

N=15

$$\boldsymbol{f(15) = \left\lfloor \frac{5(15)}{3} \right\rfloor + \sum_{i=1}^{5} \left\lceil \frac{5n - 2^i}{3} \right\rceil = \left\lceil \frac{5(15) - 2^1}{3} \right\rceil + \left\lceil \frac{5(15) - 2^2}{3} \right\rceil + \left\lceil \frac{5(15) - 2^3}{3} \right\rceil + \left\lceil \frac{5(15) - 2^4}{3} \right\rceil + \left\lceil \frac{5(15) - 2^4}{3} \right\rceil}$$
$$\boldsymbol{= 25 + 25 + 24 + 23 + 20 + 15 = 132}$$

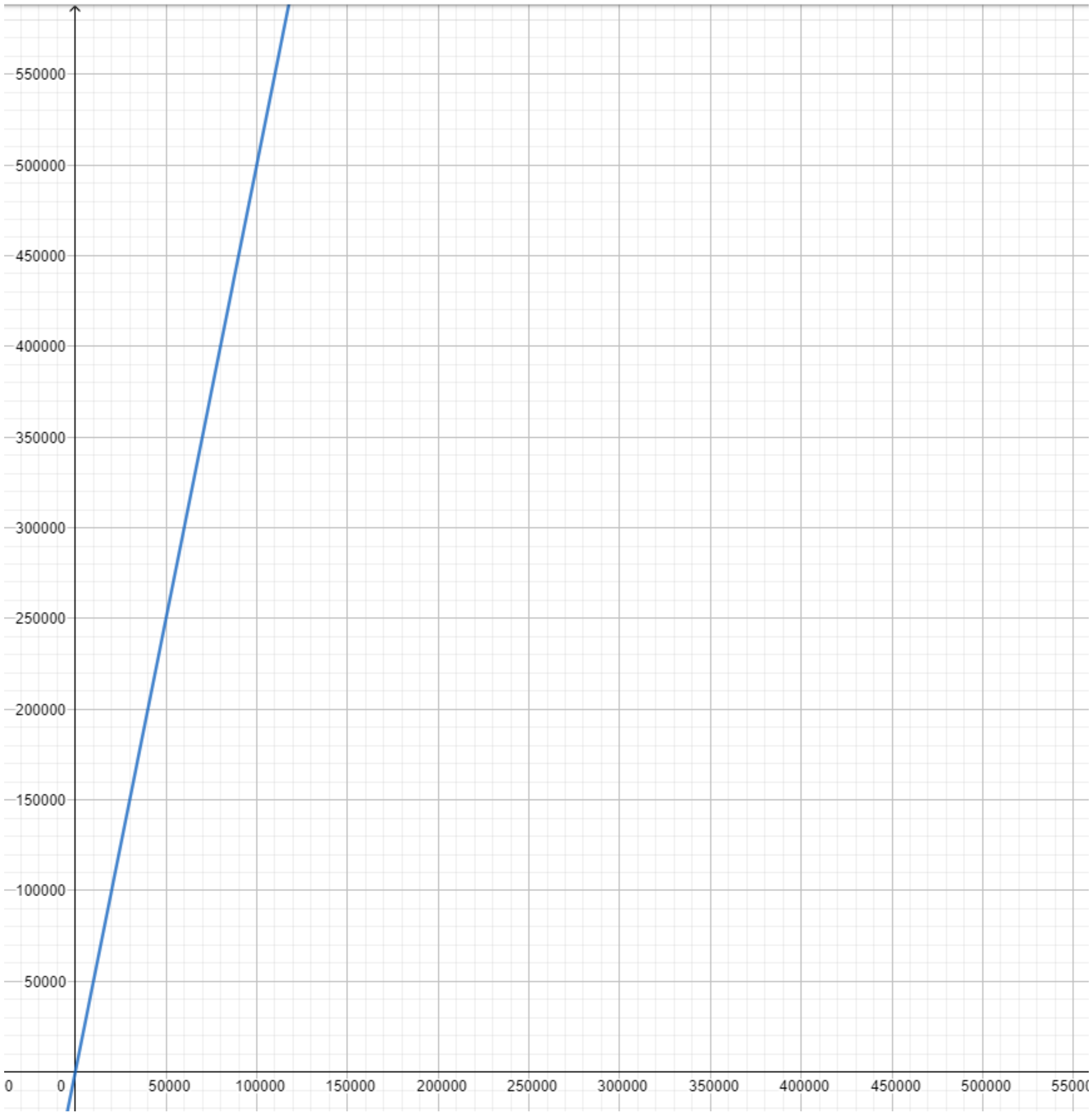
N=20

$$\boldsymbol{f(20) = \left\lfloor \frac{5(20)}{3} \right\rfloor + \sum_{i=1}^{6} \left\lceil \frac{5n - 2^i}{3} \right\rceil}$$
$$\boldsymbol{= \left\lceil \frac{5(20) - 2^1}{3} \right\rceil + \left\lceil \frac{5(20) - 2^2}{3} \right\rceil + \left\lceil \frac{5(20) - 2^3}{3} \right\rceil + \left\lceil \frac{5(20) - 2^4}{3} \right\rceil + \left\lceil \frac{5(20) - 2^4}{3} \right\rceil + \left\lceil \frac{5(20) - 2^4}{3} \right\rceil}$$
$$\boldsymbol{= 33 + 33 + 32 + 31 + 28 + 23 + 12 = 192}$$

Tabla Comparativa

N	Resultados Teóricos	Resultados Empíricos
-1	**MATH ERROR**	**0**
0	**0**	**0**
1	**3**	**3**
2	**8**	**8**
3	**17**	**17**
5	**32**	**32**
15	**132**	**132**
20	**192**	**192**
100	**1333**	**1333**
409	**6820**	**6820**
500	**8486**	**8486**
593	**10497**	**10497**
1000	**18639**	**18639**
1471	**29146**	**29146**
1500	**29776**	**29776**
2801	**59898**	**59898**
3000	**64546**	**64546**
5000	**114080**	**114080**
10000	**244827**	**244827**
20000	**522979**	**522979**

Gráfica


550000
500000
450000
400000
350000
300000
250000
200000
150000
100000
50000
0
0
50000
100000
150000
200000
250000
300000
350000
400000
450000
500000