Instituto Politécnico Nacional

Escuela Superior de Computo

Ejercicio 02

***"Calcular la Función de Complejidad"***

Mora Ayala José Antonio

Análisis de Algoritmos



Determinar para los siguientes códigos la función de complejidad temporal y espacial en términos de n. \*Considere las operaciones de: **asignación, aritméticas y condicionales.**

Determine para cada caso la función complejidad temporal y espacial (Indique detalladamente la obtención del modelo)

Contraste sus funciones con la prueba empírica (Verifique el conteo correcto de las instrucciones) para los 20 valores de

n= {−1,0,1,2,3,5,15,20,100,409,500,593,1000,1471,1500,2801,3000,5000,10000,20000}

* Incluir portada con los de datos del alumno, datos del trabajo y fotografía del alumno
* Incluir tabla comparativa de resultados teóricos vs empíricos
* Incluir la gráfica de cada modelo encontrado desde n=0 hasta n=500,000.
* Subir todo en un solo archivo de tipo documentos (Análisis, Comparativa y Códigos)
* Enmarcar los códigos en el documento y manejar formato de colores
* Recordar manejar encabezados y pies de página

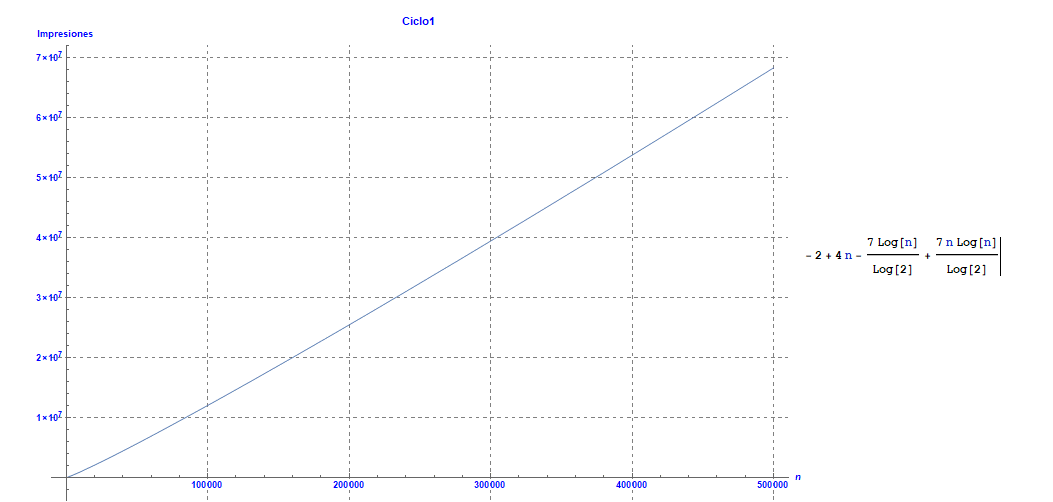
## bucle NUMERO 1

|  |
| --- |
| 1. for (i = 1, contador++; i < n; i++, contador++){ 2. contador++; //Entrando primer Bucle 3. for (j = n, contador++; j > 1; j/=2, contador++){ 4. contador++; //Entrando al Bucle 5. contador++; //temp = A[j]; 6. contador+=2; //A[j] = A[j+1]; 7. contador+=2; //A[j+1] = temp; 8. } 9. contador++; //Saliendo segundo bucle 10. } 11. contador++; //Saliendo Primer Bucle 12. printf("%d\n", contador); |

**Función de Complejidad Temporal   
­­**

**Función de Complejidad Espacial**

|  |  |  |
| --- | --- | --- |
| CICLO NUMERO 1 | | |
| n | **Empírico** | **f(n)** |
| -1 | 2 | #¡NUM! |
| 0 | 2 | #¡NUM! |
| 1 | 2 | 2 |
| 2 | 13 | 13 |
| 3 | 24 | 32.18947501 |
| 4 | 56 | 56 |
| 15 | 352 | 440.8752784 |
| 20 | 610 | 652.8164366 |
| 100 | 4556 | 5002.19234 |
| 409 | 24482 | 26412.53329 |
| 500 | 29942 | 33315.48451 |
| 593 | 39666 | 40544.06509 |
| 1000 | 66935 | 73688.7295 |
| 1471 | 108782 | 114159.364 |
| 1500 | 110928 | 116706.986 |
| 2801 | 226802 | 235655.8349 |
| 3000 | 242921 | 254482.8273 |
| 5000 | 439914 | 449981.9193 |
| 10000 | 949907 | 970044.8526 |
| 20000 | 2039900 | 2080177.719 |



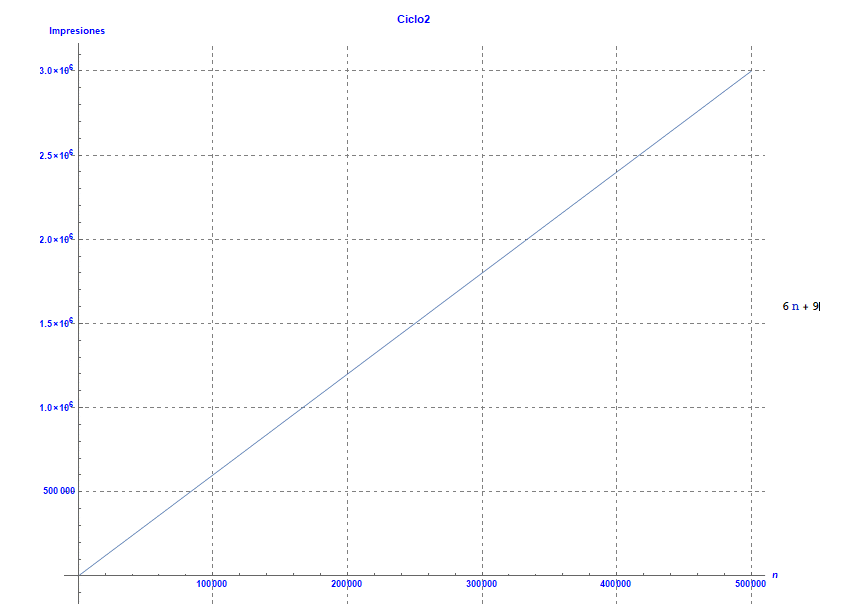
## buclE NUMERO 2

|  |
| --- |
| 1. *int* polinomio=0; 2. contador++; //Aumentando por asignacion 3. for (i = 0, contador++; i <= n; i++, contador++){ 4. contador++; //Ingresando al Bucle 5. contador+=4; //1 asignacion,3 aritmeticas 6. } 7. contador++; //Saliendo del Bucle 8. printf("%d",contador); |

|  |  |  |
| --- | --- | --- |
| CICLO NUMERO 2 | | |
| n | **Empírico** | **f(n)** |
| -1 | 3 | 3 |
| 0 | 9 | 9 |
| 1 | 15 | 15 |
| 2 | 21 | 21 |
| 3 | 27 | 27 |
| 4 | 33 | 33 |
| 15 | 99 | 99 |
| 20 | 129 | 129 |
| 100 | 609 | 609 |
| 409 | 2463 | 2463 |
| 500 | 3009 | 3009 |
| 593 | 3567 | 3567 |
| 1000 | 6009 | 6009 |
| 1471 | 8835 | 8835 |
| 1500 | 9009 | 9009 |
| 2801 | 16815 | 16815 |
| 3000 | 18009 | 18009 |
| 5000 | 30009 | 30009 |
| 10000 | 60009 | 60009 |
| 20000 | 120009 | 120009 |

**Función de Complejidad Temporal**

**Función de Complejidad Espacial**



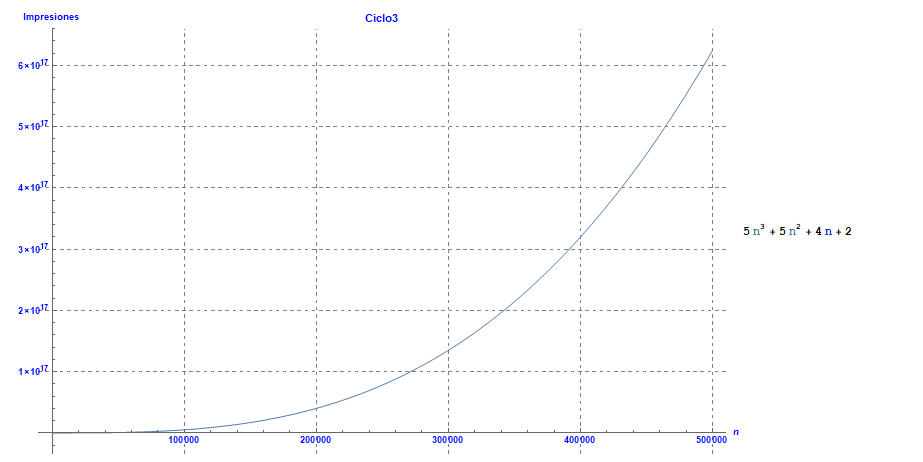
## bucle numero 3

|  |
| --- |
| 1. // Codigo 3 2. for (i = 1, contador++; i <= n; i++, contador++){ 3. contador++;                     //Entrando al primer for 4. for (j = 1, contador++; j <= n; j++, contador++){ 5. contador++;              //Entrando al segundo for 6. contador++;              //Asignacion C[i][j]=0 7. for (k = 1, contador++; k <= n; k++, contador++){ 8. contador++;        //Entrando al tercer for 9. contador+=3;  //1 Asignacion + 2 aritmeticas C[i][j] = C[i][j] + A[i][k] \* B[k][j]; 10. } 11. contador++;        //Saliendo del for k 12. } 13. contador++;                     //Saliendo for j 14. } 15. contador++;                           //Saliendo for i 16. printf("%d\n", contador); |

|  |  |  |
| --- | --- | --- |
| CICLO NUMERO 3 | | |
| n | **Empírico** | **f(n)** |
| -1 | 2 | -2 |
| 0 | 2 | 2 |
| 1 | 16 | 16 |
| 2 | 70 | 70 |
| 3 | 194 | 194 |
| 4 | 418 | 418 |
| 15 | 18062 | 18062 |
| 20 | 42082 | 42082 |
| 100 | 5050402 | 5050402 |
| 409 | 342927688 | 342927688 |
| 500 | 626252002 | 626252002 |
| 593 | 1044399904 | 1044399904 |
| 1000 | 710036706 | 5005004002 |
| 1471 | -1253993538 | 15925875646 |
| 1500 | -293613182 | 16886256002 |
| 2801 | -1752268480 | 1.09917E+11 |
| 3000 | 1901025826 | 1.35045E+11 |
| 5000 | - | 6.25125E+11 |
| 10000 | - | 5.0005E+12 |
| 20000 | - | 4.0002E+13 |
| - | El tiempo de Compilación fue excedido, por lo cual no se pudo obtener un resultado |  |

**Función de Complejidad Temporal:**

**Función de Complejidad Espacial**



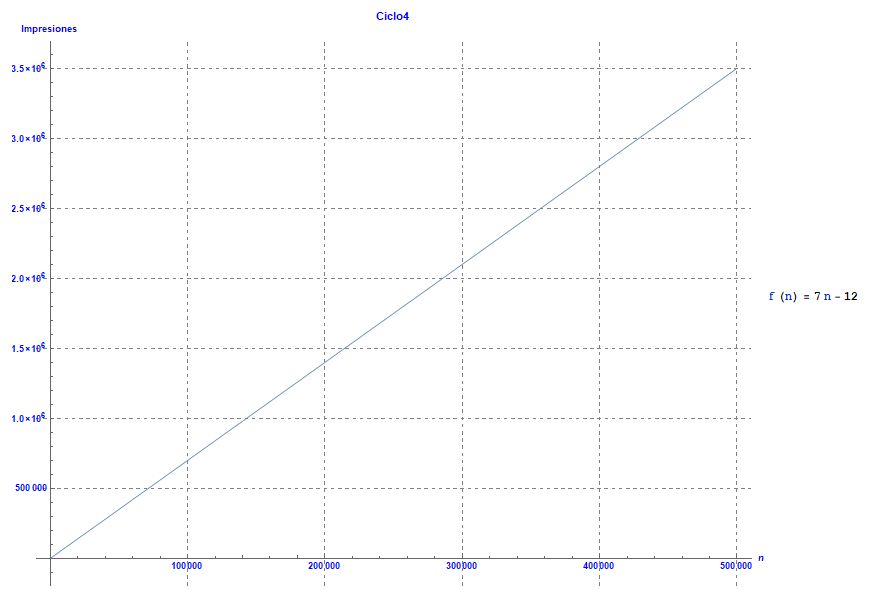
## bUCLE NUMERO 4

|  |
| --- |
| 1. // // CICLO 4 2. *int* anterior =1, actual=1,aux; 3. contador+=2;        //2 asignaciones 4. while (n>2) 5. { 6. contador++;     //Entrando al while 7. aux=anterior+actual; //Asignacion + Aritmetica 8. anterior=actual;     //Asignacion 9. actual=aux;          //Asignacion 10. n=n-1;               //Asignacion+Aritmetica 11. contador+=6; 12. } 13. printf(“%d\n”, contador); |

|  |  |  |
| --- | --- | --- |
| CICLO NUMERO 4 | | |
| n | **Empírico** | **f(n)** |
| -1 | 2 | -19 |
| 0 | 2 | -12 |
| 1 | 2 | -5 |
| 2 | 2 | 2 |
| 3 | 9 | 9 |
| 4 | 16 | 16 |
| 15 | 93 | 93 |
| 20 | 128 | 128 |
| 100 | 688 | 688 |
| 409 | 2851 | 2851 |
| 500 | 3488 | 3488 |
| 593 | 4139 | 4139 |
| 1000 | 6988 | 6988 |
| 1471 | 10285 | 10285 |
| 1500 | 10488 | 10488 |
| 2801 | 19595 | 19595 |
| 3000 | 20988 | 20988 |
| 5000 | 34988 | 34988 |
| 10000 | 69988 | 69988 |
| 20000 | 139988 | 139988 |

**Función de Complejidad Temporal:**

**Función de Complejidad Espacial**

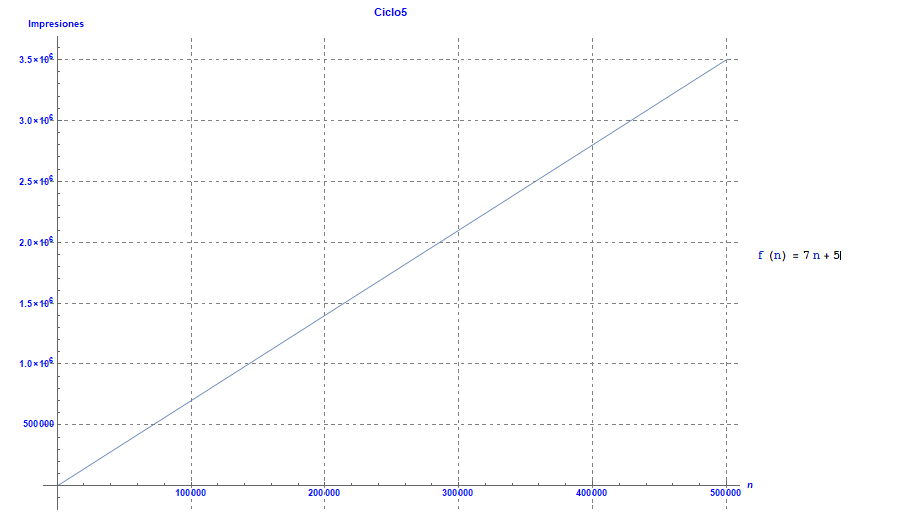
****

## bUCLE NUMERO 5

|  |
| --- |
| 1. // CICLO 5 2. for (i = n - 1, j = 0, contador+=2; i >= 0; i--, j++, contador+=2) 3. { 4. contador++; //Ingresando al for 5. s2[j]=s[i]; 6. contador++; //Asignacion 7. } 8. contador++; //saliendo for 9. for (k = 0, contador++; k < n; k++, contador++){ 10. contador++; //Entrando al for 11. s[i]=s2[j]; 12. contador++; //Asignacion 13. } 14. contador++; //Saliendo for 15. printf("%d\n", contador); |

|  |  |  |
| --- | --- | --- |
| CICLO NUMERO 5 | | |
| n | **Empírico** | **f(n)** |
| -1 | 5 | -2 |
| 0 | 5 | 5 |
| 1 | 12 | 12 |
| 2 | 19 | 19 |
| 3 | 26 | 26 |
| 4 | 33 | 33 |
| 15 | 110 | 110 |
| 20 | 145 | 145 |
| 100 | 705 | 705 |
| 409 | 2868 | 2868 |
| 500 | 3505 | 3505 |
| 593 | 4156 | 4156 |
| 1000 | 7005 | 7005 |
| 1471 | 10302 | 10302 |
| 1500 | 10505 | 10505 |
| 2801 | 19612 | 19612 |
| 3000 | 21005 | 21005 |
| 5000 | 35005 | 35005 |
| 10000 | 70005 | 70005 |
| 20000 | 140005 | 140005 |

**Función de Complejidad Temporal**

**Función de Complejidad Espacial**

## Bucle numero 6

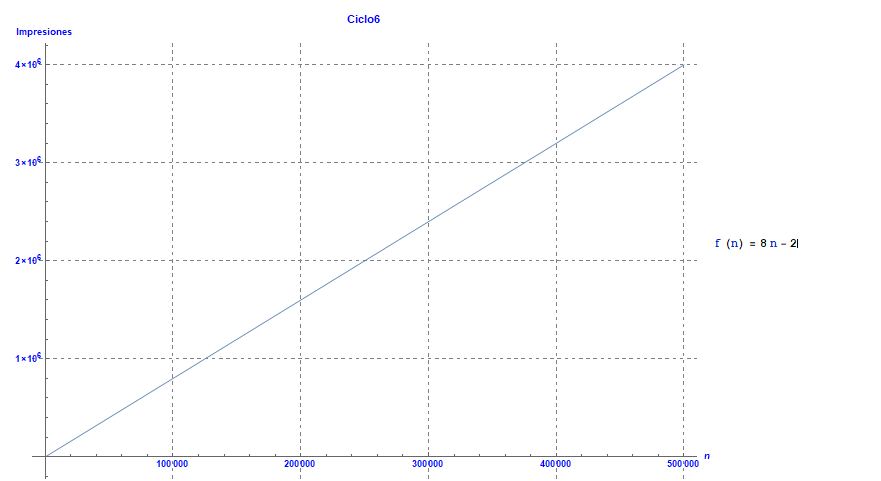
|  |
| --- |
| 1. // CICLO 6 2. *int* l,a=0,b=0,r; 3. l=(a<b)?a:b; 4. contador+=3; //2 Comparaciones + 1 asignacion 5. r=1; 6. contador++; //Asignacion 7. for (i = 2, contador++; i <= n; i++,contador++) 8. { 9. contador++; //Entrando al Bucle 10. if (a%i==0 && b%i==0) 11. { 12. contador+=5; //2 aritmeticas, 1 asignacion,3 comparaciones 13. r=i; 14. contador++; //1 asignacion 15. } 16. } 17. contador++; //Saliendo del for 18. printf("%d\n", contador); |

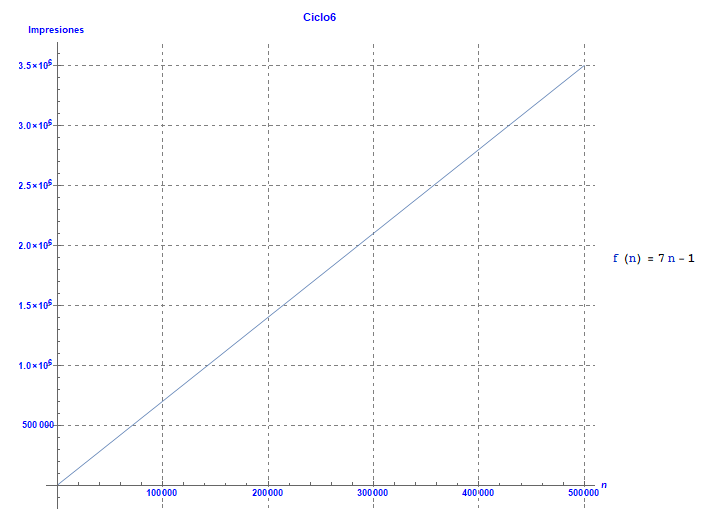
|  |  |  |
| --- | --- | --- |
| CICLO NUMERO 6 (Cumpliendo) | | |
| n | **Empírico** | **f(n)** |
| -1 | 6 | -10 |
| 0 | 6 | -2 |
| 1 | 6 | 6 |
| 2 | 14 | 14 |
| 3 | 22 | 22 |
| 4 | 30 | 30 |
| 15 | 118 | 118 |
| 20 | 158 | 158 |
| 100 | 798 | 798 |
| 409 | 3270 | 3270 |
| 500 | 3998 | 3998 |
| 593 | 4742 | 4742 |
| 1000 | 7998 | 7998 |
| 1471 | 11766 | 11766 |
| 1500 | 11998 | 11998 |
| 2801 | 22406 | 22406 |
| 3000 | 23998 | 23998 |
| 5000 | 39998 | 39998 |
| 10000 | 79998 | 79998 |
| 20000 | 159998 | 159998 |

**Función de Complejidad Temporal**

**Función de Complejidad Espacial**

|  |  |  |
| --- | --- | --- |
| CICLO NUMERO 6 (No cumpliendo) | | |
| n | **Empírico** | **f(n)** |
| -1 | 6 | -8 |
| 0 | 6 | -1 |
| 1 | 6 | 6 |
| 2 | 13 | 13 |
| 3 | 20 | 20 |
| 4 | 27 | 27 |
| 15 | 104 | 104 |
| 20 | 139 | 139 |
| 100 | 699 | 699 |
| 409 | 2862 | 2862 |
| 500 | 3499 | 3499 |
| 593 | 4150 | 4150 |
| 1000 | 6999 | 6999 |
| 1471 | 10296 | 10296 |
| 1500 | 10499 | 10499 |
| 2801 | 19606 | 19606 |
| 3000 | 20999 | 20999 |
| 5000 | 34999 | 34999 |
| 10000 | 69999 | 69999 |
| 20000 | 139999 | 139999 |





## Ciclo numero 7

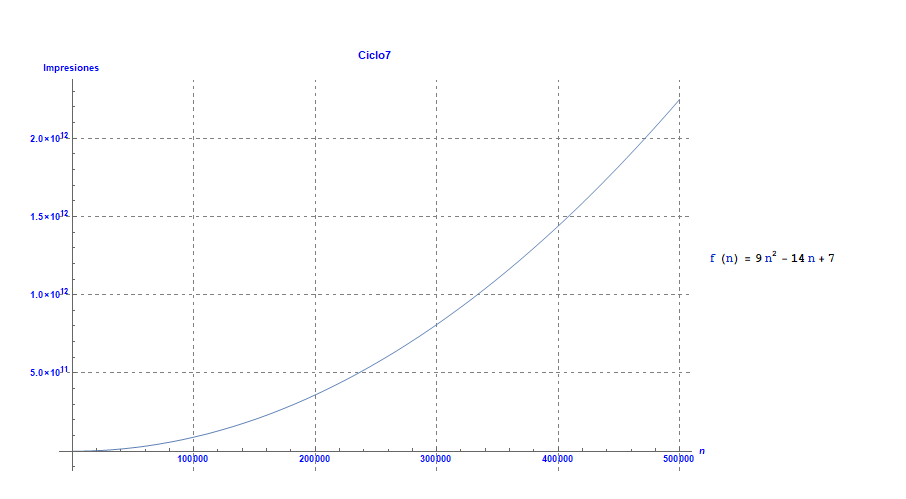
|  |
| --- |
| 1. // CICLO 7 2. for (i = 1,contador++; i < n;contador++, i++){ 3. contador++;                          //Entrando al for 4. for (j = 0,contador++; j < n-1;contador++, j++) 5. { 6. contador++;             //Entrando al for j 7. if (lista[j]>lista[j+1]) 8. { 9. contador+=2      //1 comparacion + 1 aritmetica 10. temp=lista[j];  //Asignacion 11. lista[j]=lista[j+1]; //1 asigancion+1aritmetica 12. lista[j+1]=temp;  //1 Aritmetica + 1 asignacion 13. contador+=5 14. } 15. } 16. contador++;  //Saliendo del for 17. } 18. contador++;  //Saliendo del for 19. printf("%d\n", contador); |

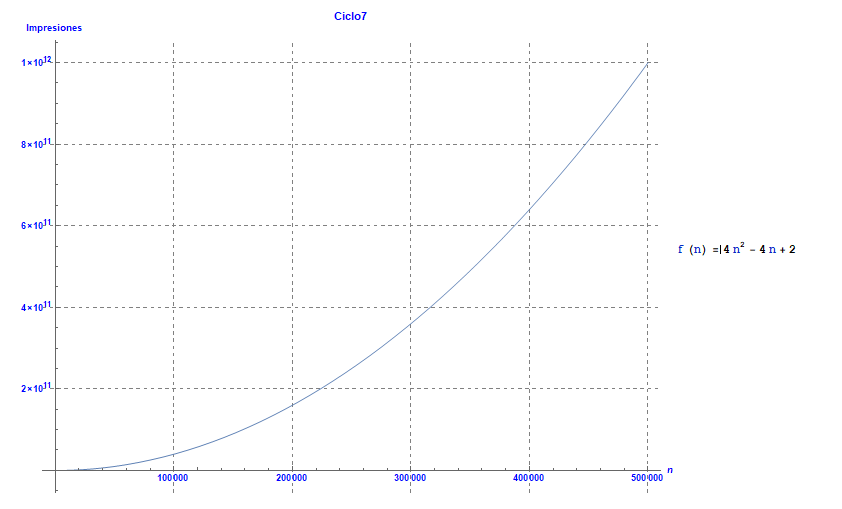
|  |  |  |
| --- | --- | --- |
| CICLO NUMERO 7 (Cumpliendo) | | |
| n | **Empírico** | **f(n)** |
| -1 | 2 | 30 |
| 0 | 2 | 7 |
| 1 | 2 | 2 |
| 2 | 15 | 15 |
| 3 | 46 | 46 |
| 4 | 95 | 95 |
| 15 | 1822 | 1822 |
| 20 | 3327 | 3327 |
| 100 | 88607 | 88607 |
| 409 | 1499810 | 1499810 |
| 500 | 2243007 | 2243007 |
| 593 | 3156546 | 3156546 |
| 1000 | 8986007 | 8986007 |
| 1471 | 19453982 | 19453982 |
| 1500 | 20229007 | 20229007 |
| 2801 | 70571202 | 70571202 |
| 3000 | 80958007 | 80958007 |
| 5000 | 224930007 | 224930007 |
| 10000 | 899860007 | 899860007 |
| 20000 | -695247289 | 3599720007 |

|  |  |  |
| --- | --- | --- |
| CICLO NUMERO 7 (No cumpliendo) | | |
| n | **Empírico** | **f(n)** |
| -1 | 2 | 10 |
| 0 | 2 | 2 |
| 1 | 2 | 2 |
| 2 | 10 | 10 |
| 3 | 26 | 26 |
| 4 | 50 | 50 |
| 15 | 842 | 842 |
| 20 | 1522 | 1522 |
| 100 | 39602 | 39602 |
| 409 | 667490 | 667490 |
| 500 | 998002 | 998002 |
| 593 | 1404226 | 1404226 |
| 1000 | 3996002 | 3996002 |
| 1471 | 8649482 | 8649482 |
| 1500 | 8994002 | 8994002 |
| 2801 | 31371202 | 31371202 |
| 3000 | 35988002 | 35988002 |
| 5000 | 99980002 | 99980002 |
| 10000 | 399960002 | 399960002 |
| 20000 | 1599920002 | 1599920002 |

**Función de Complejidad Temporal**

**Función de complejidad Espacial**





## Procedimiento realizado a mano

