

PRÁCTICA 1: DETERMINACIÓN EXPERIMENTAL DE LA COMPLEJIDAD TEMPORAL DE UN ALGORITMO

Mendoza Martínez Eduardo, Nombre Completo Alumno 2.

Escuela Superior de Cómputo
Instituto Politécnico Nacional, México
edoomm8@gmail.com, correo@alumno2

Resumen: Redactar de manera breve y concisa de que trata el trabajo presentado. Un sólo párrafo. **Palabras Clave:** Colocar de 3 a 5 palabras clave.

1 Introducción

En esta sección, como su nombre lo indica, introducir al lector al trabajo presentado. En el caso de esta primera práctica podrían comenzar explicando la importancia de los algoritmos y la importancia de analizar un algoritmo, y finalizar con el objetivo de la práctica. Indicaciones para el envío de la práctica: 1) Tienen a los más 8 días para enviar sus prácticas a la cuenta de correo que se les proporcionara. 2) En "Asunto" del correo escribir el número de práctica (Práctica No. 1), y en el contenido del correo escribir los nombres completos de los integrantes del equipo (2 personas a lo más) y si tienen algún comentario respecto a la práctica. 3) En cada Práctica se tiene que enviar todo el código fuente de sus programas y en el contenido del correo especificar en que lenguajes fueron realizados. Además enviar el reporte en PDF y su respectivo código en LaTeX. 4) Recuerden que los códigos de sus algoritmos implementados deben de tener plantilla de datos y comentarios.

2 Conceptos Básicos

Aquí va todo lo necesario para comprender el trabajo. En el caso de esta práctica podrían colocar los conceptos de Θ , O y Ω . Además comentar que algoritmos se desarrollarán y mostrar los algoritmos, pueden mostrar ejemplos del funcionamiento del algoritmo implementado.

3 Experimentación y Resultados

3.1 Suma binaria

Para esta parte de la práctica fue necesario implementar un algoritmo capaz de realizar una suma binaria entre dos términos A y B. El algoritmo se muestra a continuación en pseudocódigo.

```

suma (A[1, ..., n], B[1, ..., n])
  i <- n - 1
  acarreo <- 0
  C[n]

  while i != -1 do
    if acarreo == 0 then
      if A[i] == 0 and B[i] == 0 then
        acarreo <- 0
        c <- 0
      else if A[i] == 1 and B[i] == 1 then
        acarreo <- 1
        c <- 0
      else
        acarreo <- 0
        c <- 1
    else
      if A[i] == 0 and B[i] == 0 then
        acarreo <- 0
        c <- 1
      else if A[i] == 1 and B[i] == 1 then
        acarreo <- 1
        c <- 1
      else
        acarreo <- 1
        c <- 1

    C[0] <- c
    i <- i - 1

```

```

if acarreo == 1 then
  C[0] <- acarreo

```

Las pruebas de escritorio se implementaron en Python y a continuación se muestran algunas pruebas con números binarios aleatorios de número de bits no requeridos, esto con el fin de verificar si las sumas las hace correctamente.

```

C:\Windows\System32\cmd.exe

C:\Users\52556\Documents\ESCOM\Análisis de algoritmos\P1>py suma.py
[1]
+
[0]
[1]

[1, 0]
+
[1, 1]
[1, 0, 1]

[1, 0, 1]
+
[1, 1, 1]
[1, 1, 0, 0]

[0, 1, 0, 0]
+
[0, 1, 0, 0]
[1, 0, 0, 0]

[0, 1, 0, 1, 1]
+
[1, 1, 1, 1, 0]
[1, 0, 1, 0, 0, 1]

[1, 0, 1, 1, 1, 0]
+
[1, 1, 1, 0, 1, 0]
[1, 1, 0, 1, 0, 0, 0]

[1, 0, 1, 1, 0, 1, 0]
+
[1, 0, 1, 1, 0, 0, 0]
[1, 0, 1, 1, 0, 0, 1, 0]

```

Una vez verificado los resultados, procedimos a cambiar el número de bits a los acordados, es decir, a potencias de 2, y en vez de mostrar la suma, mostramos el tiempo de ejecución, como se muestra a continuación.

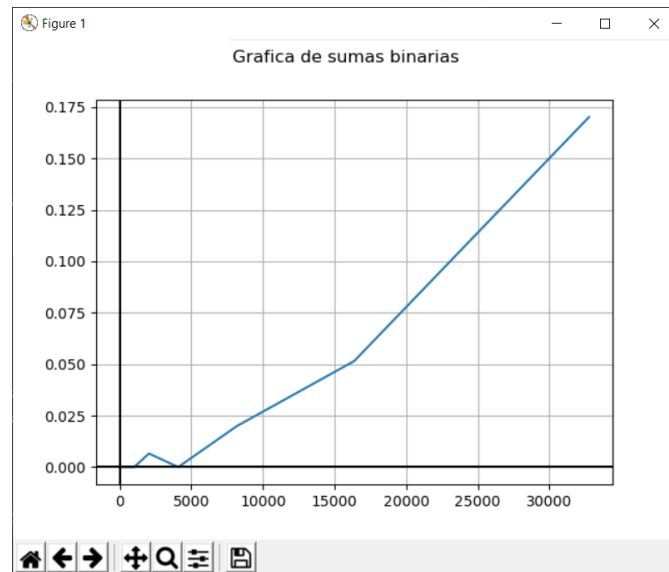
```

Select C:\Windows\System32\cmd.exe - py suma.py

C:\Users\52556\Documents\ESCOM\Análisis de algoritmos\P1>py suma.py
1 -- Suma con dos terminos de 2 bits, tiempo: 0.0 segundos --
2 -- Suma con dos terminos de 4 bits, tiempo: 0.0 segundos --
3 -- Suma con dos terminos de 8 bits, tiempo: 0.0 segundos --
4 -- Suma con dos terminos de 16 bits, tiempo: 0.0 segundos --
5 -- Suma con dos terminos de 32 bits, tiempo: 0.0 segundos --
6 -- Suma con dos terminos de 64 bits, tiempo: 0.0 segundos --
7 -- Suma con dos terminos de 128 bits, tiempo: 0.0 segundos --
8 -- Suma con dos terminos de 256 bits, tiempo: 0.0 segundos --
9 -- Suma con dos terminos de 512 bits, tiempo: 0.0 segundos --
10 -- Suma con dos terminos de 1024 bits, tiempo: 0.0 segundos --
11 -- Suma con dos terminos de 2048 bits, tiempo: 0.006578683853149414 segundos --
12 -- Suma con dos terminos de 4096 bits, tiempo: 0.0 segundos --
13 -- Suma con dos terminos de 8192 bits, tiempo: 0.019946575164794922 segundos --
14 -- Suma con dos terminos de 16384 bits, tiempo: 0.051573753356933594 segundos --
15 -- Suma con dos terminos de 32768 bits, tiempo: 0.17020225524902344 segundos --

```

La gráfica de estos primeros tiempos quedo conformada de la siguiente forma

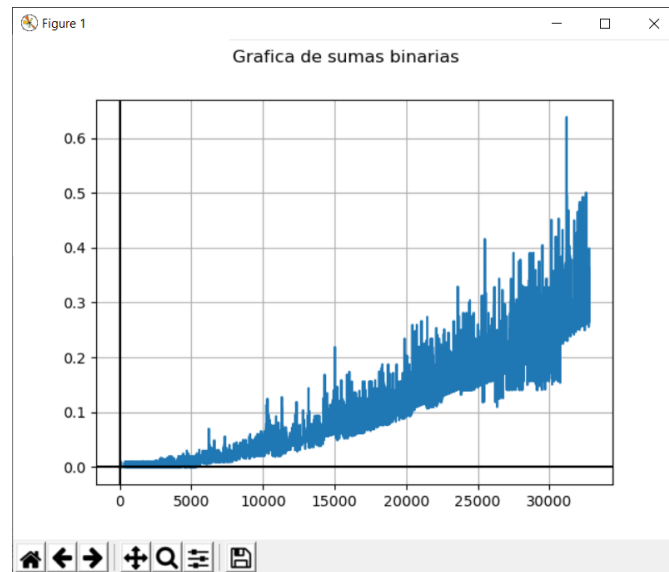


Este mismo experimento se realizo incrementando el número de bits uno a uno, tal que fue posible observar la forma de la curva que va tomando el algoritmo implementado.

```

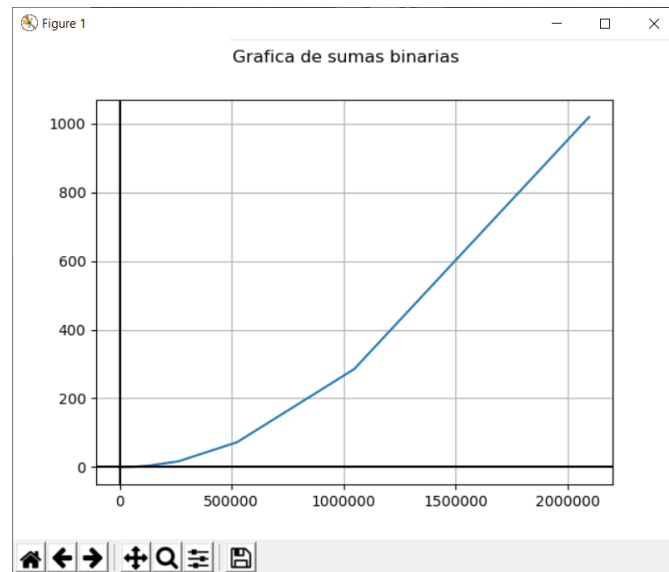
C:\Windows\System32\cmd.exe - py sumapy
32747 -- Suma con dos terminos de 32747 bits, tiempo: 0.2553746700286805 segundos --
32748 -- Suma con dos terminos de 32748 bits, tiempo: 0.34366822242736816 segundos --
32749 -- Suma con dos terminos de 32749 bits, tiempo: 0.28147053718566895 segundos --
32750 -- Suma con dos terminos de 32750 bits, tiempo: 0.2652900218963623 segundos --
32751 -- Suma con dos terminos de 32751 bits, tiempo: 0.26625061031515625 segundos --
32752 -- Suma con dos terminos de 32752 bits, tiempo: 0.363631010055542 segundos --
32753 -- Suma con dos terminos de 32753 bits, tiempo: 0.2655925750732422 segundos --
32754 -- Suma con dos terminos de 32754 bits, tiempo: 0.26529455184936523 segundos --
32755 -- Suma con dos terminos de 32755 bits, tiempo: 0.2652904987335205 segundos --
32756 -- Suma con dos terminos de 32756 bits, tiempo: 0.2652904987335205 segundos --
32757 -- Suma con dos terminos de 32757 bits, tiempo: 0.2652924060821533 segundos --
32758 -- Suma con dos terminos de 32758 bits, tiempo: 0.26429009437561035 segundos --
32759 -- Suma con dos terminos de 32759 bits, tiempo: 0.26628684997558594 segundos --
32760 -- Suma con dos terminos de 32760 bits, tiempo: 0.2652897034777832 segundos --
32761 -- Suma con dos terminos de 32761 bits, tiempo: 0.26628708839416504 segundos --
32762 -- Suma con dos terminos de 32762 bits, tiempo: 0.3989331722295215 segundos --
32763 -- Suma con dos terminos de 32763 bits, tiempo: 0.26529383659362793 segundos --
32764 -- Suma con dos terminos de 32764 bits, tiempo: 0.2652900218963623 segundos --
32765 -- Suma con dos terminos de 32765 bits, tiempo: 0.26528120040893555 segundos --
32766 -- Suma con dos terminos de 32766 bits, tiempo: 0.2652888298034668 segundos --
32767 -- Suma con dos terminos de 32767 bits, tiempo: 0.26628732681274414 segundos --

```



Y se repitió el experimento con potencias de 2, primero se muestra con 2^{21} bits con un tiempo tardado de 17 minutos aproximadamente

```
C:\Windows\System32\cmd.exe - py sumapy
C:\Users\S2556\Documents\ESCOM\Análisis de algoritmos\PI>py suma.py
1 -- Suma con dos terminos de 2 bits, tiempo: 0.0 segundos --
2 -- Suma con dos terminos de 4 bits, tiempo: 0.0 segundos --
3 -- Suma con dos terminos de 8 bits, tiempo: 0.0 segundos --
4 -- Suma con dos terminos de 16 bits, tiempo: 0.0 segundos --
5 -- Suma con dos terminos de 32 bits, tiempo: 0.0 segundos --
6 -- Suma con dos terminos de 64 bits, tiempo: 0.0 segundos --
7 -- Suma con dos terminos de 128 bits, tiempo: 0.0 segundos --
8 -- Suma con dos terminos de 256 bits, tiempo: 0.0 segundos --
9 -- Suma con dos terminos de 512 bits, tiempo: 0.0 segundos --
10 -- Suma con dos terminos de 1024 bits, tiempo: 0.0 segundos --
11 -- Suma con dos terminos de 2048 bits, tiempo: 0.0009968280792236328 segundos --
12 -- Suma con dos terminos de 4096 bits, tiempo: 0.003989458884106445 segundos --
13 -- Suma con dos terminos de 8192 bits, tiempo: 0.014960289801464884 segundos --
14 -- Suma con dos terminos de 16384 bits, tiempo: 0.04886817932128986 segundos --
15 -- Suma con dos terminos de 32768 bits, tiempo: 0.1845107078552246 segundos --
16 -- Suma con dos terminos de 65536 bits, tiempo: 0.8127932548522949 segundos --
17 -- Suma con dos terminos de 131072 bits, tiempo: 4.1483893394470215 segundos --
18 -- Suma con dos terminos de 262144 bits, tiempo: 16.53691029548645 segundos --
19 -- Suma con dos terminos de 524288 bits, tiempo: 72.21686922149658 segundos --
20 -- Suma con dos terminos de 1048576 bits, tiempo: 285.54307746887207 segundos --
21 -- Suma con dos terminos de 2097152 bits, tiempo: 1019.3919959068298 segundos --
```



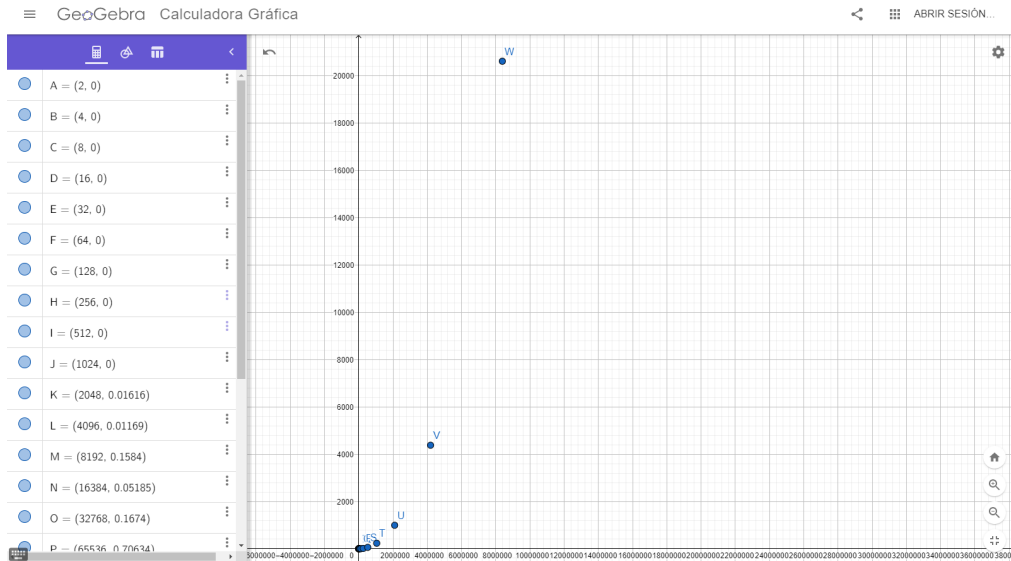
Y por último se muestra con 2^{23} bits, donde los últimos puntos de la gráfica se vieron un poco afectados por la computadora que estuvo realizando otras tareas, ya que para este proceso se tardó en ejecutar un aproximado de 5 horas y media

```

C:\Windows\System32\cmd.exe

2 -- Suma con dos terminos de 4 bits, tiempo: 0.0 segundos --
3 -- Suma con dos terminos de 8 bits, tiempo: 0.0 segundos --
4 -- Suma con dos terminos de 16 bits, tiempo: 0.0 segundos --
5 -- Suma con dos terminos de 32 bits, tiempo: 0.0 segundos --
6 -- Suma con dos terminos de 64 bits, tiempo: 0.0 segundos --
7 -- Suma con dos terminos de 128 bits, tiempo: 0.0 segundos --
8 -- Suma con dos terminos de 256 bits, tiempo: 0.0 segundos --
9 -- Suma con dos terminos de 512 bits, tiempo: 0.0 segundos --
10 -- Suma con dos terminos de 1024 bits, tiempo: 0.0 segundos --
11 -- Suma con dos terminos de 2048 bits, tiempo: 0.01616048812866211 segundos --
12 -- Suma con dos terminos de 4096 bits, tiempo: 0.011699914932250977 segundos --
13 -- Suma con dos terminos de 8192 bits, tiempo: 0.01584005355834961 segundos --
14 -- Suma con dos terminos de 16384 bits, tiempo: 0.051850318908691406 segundos --
15 -- Suma con dos terminos de 32768 bits, tiempo: 0.16740965843200684 segundos --
16 -- Suma con dos terminos de 65536 bits, tiempo: 0.706343412399292 segundos --
17 -- Suma con dos terminos de 131072 bits, tiempo: 3.812422037124634 segundos --
18 -- Suma con dos terminos de 262144 bits, tiempo: 13.440423727035522 segundos --
19 -- Suma con dos terminos de 524288 bits, tiempo: 57.93163728713989 segundos --
20 -- Suma con dos terminos de 1048576 bits, tiempo: 231.87008452415466 segundos --
21 -- Suma con dos terminos de 2097152 bits, tiempo: 991.9913923740387 segundos --
22 -- Suma con dos terminos de 4194304 bits, tiempo: 4375.415351629257 segundos --
23 -- Suma con dos terminos de 8388608 bits, tiempo: 20607.315758228382 segundos --

```

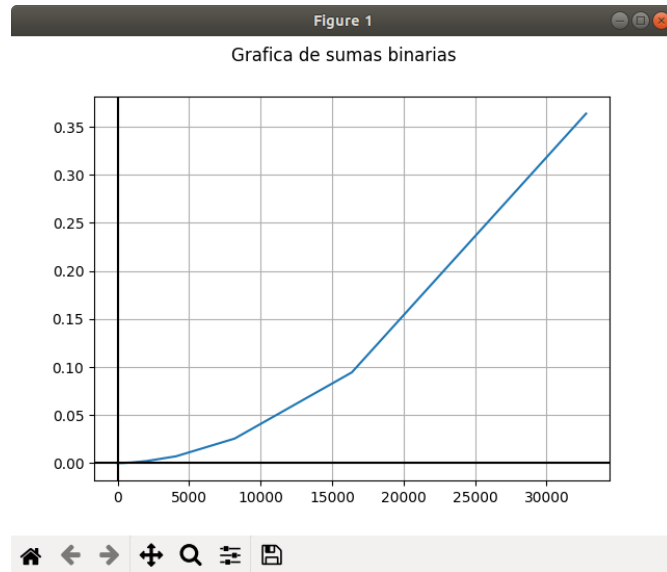


Para proponer una función $O(g(n))$ observamos que si se realiza esta prueba en otra computadora, los tiempos pueden variar muy poco

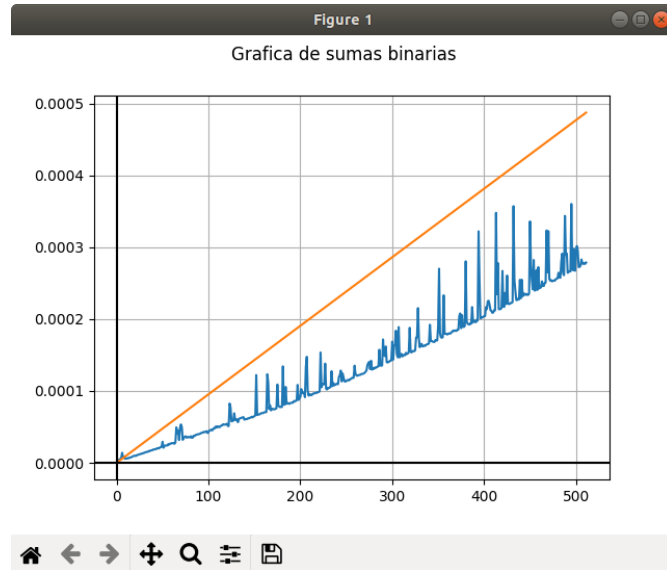
```

File Edit View Search Terminal Help
Pygments, readme-renderer, pyparser, cffi, cryptography, jeepney, SecretStorage, keyring, twine, SimpleWebSocketServer, pyloco, matplotlib
Successfully installed Pygments-2.5.2 SecretStorage-3.1.2 SimpleWebSocketServer-0.1.1 bleach-3.1.0 certifi-2019.11.28 cffi-1.13.2 chardet-3.0.4
cryptography-2.8 cyclr-0.10.0 docutils-0.16 idna-2.8 importlib-metadata-1.5.0 jeepney-0.4.2 keyring-21.1.0 kiwisolver-1.1.0 matplotlib-0.1.9 m
atplotlib-3.1.3 numpy-1.18.1 pkginfo-1.5.0.1 pyparser-2.19 pyloco-0.0.139 pyparsing-2.4.6 python-dateutil-2.8.1 readme-renderer-24.0 requests
2.22.0 requests-toolbelt-0.9.1 setuptools-45.1.0 six-1.14.0 tqdm-4.42.0 twine-3.1.1 typing-3.7.4.1 urllib3-1.25.8 ushlex-0.99.1 webencodings-
0.5.1 websocket-client-0.57.0 zipp-2.1.0
edoomm@Satellite-L735:~/Documents/ESCOM$ python3 suma.py
1 -- Suma con dos terminos de 2 bits, tiempo: 6.4373016357421875e-06 segundos --
2 -- Suma con dos terminos de 4 bits, tiempo: 7.152557373046875e-06 segundos --
3 -- Suma con dos terminos de 8 bits, tiempo: 8.821487426757812e-06 segundos --
4 -- Suma con dos terminos de 16 bits, tiempo: 1.5020370483398438e-05 segundos --
5 -- Suma con dos terminos de 32 bits, tiempo: 2.5272369384765625e-05 segundos --
6 -- Suma con dos terminos de 64 bits, tiempo: 4.887580871582031e-05 segundos --
7 -- Suma con dos terminos de 128 bits, tiempo: 0.00010132789611816406 segundos --
8 -- Suma con dos terminos de 256 bits, tiempo: 0.00011801719665527344 segundos --
9 -- Suma con dos terminos de 512 bits, tiempo: 0.0002810955047607422 segundos --
10 -- Suma con dos terminos de 1024 bits, tiempo: 0.0007350444793701172 segundos --
11 -- Suma con dos terminos de 2048 bits, tiempo: 0.0022237300872802734 segundos --
12 -- Suma con dos terminos de 4096 bits, tiempo: 0.007136344909667969 segundos --
13 -- Suma con dos terminos de 8192 bits, tiempo: 0.025573253631591797 segundos --
14 -- Suma con dos terminos de 16384 bits, tiempo: 0.09441065788269043 segundos --
15 -- Suma con dos terminos de 32768 bits, tiempo: 0.36365771293640137 segundos --

```



Por lo tanto, si este algoritmo es analizado a priori, podemos observar que tendra $O(n)$, esto de igual forma se verifica con el analisis a posteriori realizado en la practica y lo podemos ver claramente en la siguiente gráfica



Así también, se puede acotar por otra función $c_1g(n)$, pero en la notación O los coeficientes pueden ser omitidos Así bien,

$$Suma \in O(n)$$

4 Conclusiones

Las conclusiones de manera general y de manera individual. En lo general, podrían escribir, errores que se presentaron y como se resolvieron, podrían

escribir observaciones de como mejorar el algoritmo, si quedaron los resultados esperados o no, y por que, etc. Las conclusiones individuales ya cada quien sabrá que redactar. A las conclusiones individuales se les anexará una fotografía del cada alumno. Conclusiones Alumno 1 (FOTO) Conclusiones ALumno 2 (FOTO)

5 Anexo

Si se dejan problemas para resolver, en esta sección se mostrarán.

6 Bibliografía

Mostrar referencias en formato IEEE.