

Instituto Politécnico Nacional

Escuela Superior de Cómputo

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

Tejeda Moyao Leon Francisco
leontejeda@gmail.com

Resumen:

En la presente practica, la tarea fue desarrollar 2 algortimos.

- Encontrar un valor repetido en un arreglo.
- Elaborar el algoritmo de euclides.

Palabras Clave: Complejidad, Algoritmo, C

Introducción.

- **Importancia de de los algortimos.**

Los algoritmos se pueden definir como una sucesión de pasos que se deben de seguir para desarrollar una tarea dada.

Por lo tanto se podría decir que la importancia de los algoritmos radica en que al momento de deifnir un algoritmo, este puede implemnmetarse en uno o varios problemas y, de cierta manera, es mas facil resolver ciertos problemas.

- **Importancia de analizar un algoritmo.**

Al momento de analizar un algoritmo, se puede llegar a saber el número de pasos que realiza y/o el número de tiempo que le toma llegar al resultado deseado.

NOTA: Que el codigo del algoritmo sea mas pequeño, no significa que tarde menos tiempo)

Independientemente de lo que se requira analizar, analizando el algoritmo se puede llegar a saber en que parte el algoritmo puede moejorar para asi ahorrar recursos, ya sea en tiempo o memoria.

- **Objetivo.**

Realizar el analisis a postereori de ambos codigos realizados y ver sus graficas en el programa demos.

Conceptos Básicos.

- **Algortimo.**

Secuencia de pasos o insturcciones, meidante las cuales se llegar a realizar una tarea.

- **Complejidad Algoritmica.**

Se traduce como la descripción de un algoritmo en cuanto a tiempo de ejecución y memoria requerida.

- **Analisis Apostereori.**

Es el anilisis de un algortimo despues de que este ha sido ejecutado, recolectando la información obtenida durante su ejecución.

Experimentación y Resultados.

Para el primer algoritmo, se encargó que en un arreglo de tamaño N se llenara con valores aleatorios que fueran desde 1 hasta $3N$, posteriormente este arreglo se dividiría en 2 partes. Finalmente se buscará un valor repetido en ambas mitades del arreglo, cuando encuentre un valor repetido el programa finalizará.

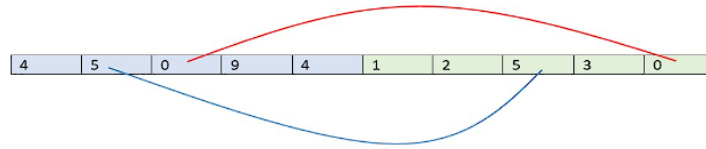


Figura 1: Ejemplo arreglo.

Habiendo corrido el algoritmo, este imprime los valores obtenidos, indica los valores ingresados y termina indicando si encontró el valor y en qué posición.

```
--N| mero: 1100 ingresado --
--N| mero: 1945 ingresado --
--N| mero: 340 ingresado --
--N| mero: 215 ingresado --
--N| mero: 2010 ingresado --
--N| mero: 1657 ingresado --
--N| mero: 2548 ingresado --
--N| mero: 1983 ingresado --
--N| mero: 173 ingresado --
--N| mero: 1908 ingresado --
--N| mero: 2256 ingresado --
--N| mero: 1716 ingresado --
--N| mero: 707 ingresado --
--N| mero: 697 ingresado --
--N| mero: 510 ingresado --
--N| mero: 1955 ingresado --
--N| mero: 1308 ingresado --
--N| mero: 251 ingresado --
--N| mero: 2506 ingresado --
--N| mero: 1146 ingresado --
--N| mero: 2286 ingresado --
--N| mero: 1363 ingresado --
--N| mero: 774 ingresado --
Dato: 845 encontrado en posiciones: 5 y 877
N|| mero de pasos: 10913
```

Figura 2: Funcionamiento primer algoritmo.

El número de pasos y el tamaño del arreglo son guardados en un archivos con extensión .csv. En la primera columna será el tamaño y en la segunda el número de pasos.

10	137
20	197
30	428
40	370
50	892
60	736
70	589
80	1536
90	1897
100	975
110	2107

Figura 3: Datos programa 1.

Se hizo uso del programa desmos, el cual permite mostrar la gráfica obtenida, esta se obtiene pegando los datos del archivo .csv.

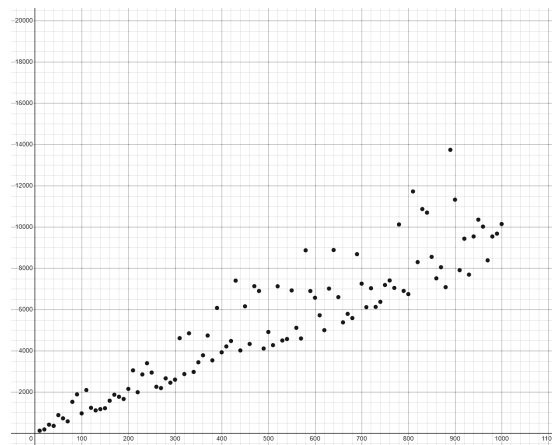


Figura 4: Gráfica programa 1.

Se buscó tanto la cota superior como la cota inferior y se procedió a comprobar si la función estaba acotada por dichas funciones.

- Cota Superior: $\Omega(1)$
- Cota Inferior: $\mathcal{O}(N^2)$

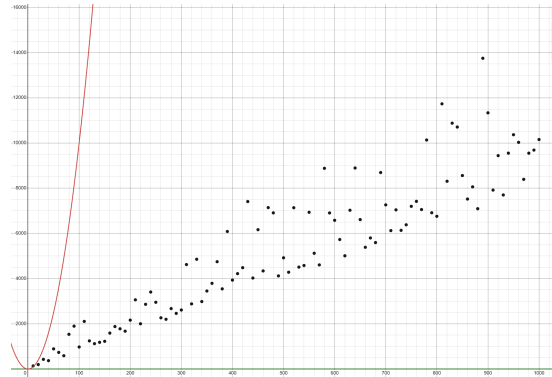


Figura 5: Gráfica programa 1 con cotas.

Conclusiones.

En esta practica se usaron herramientas que ayudaron a la interpretación del comportamiento del código, esto gracias a los datos que fueron guardados y almacenados en los archivos con extensión .csv.

Durante el desarrollo de esta se pudo observar que en el primer caso se podia mejorar el algortimo, al solo recorrer la mitad del arreglo en los ciclos implementados en vez de recorrer todo el arreglo.



Conclusiones Alumno 1:

En la practica se pudo visualizar el comportamiento del algortimos que fue utilizado, esto nos ayudó a poder visualizar una mejor manera de ocmo realizar el mismo trabajo con menor tiempo simplemente modificando el algortimo solo un poco.