

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

Tejeda Moyao León Francisco.

leontejeda@gmail.com

Resumen: Este programa realiza una comparación de datos entre subarreglos de tamaño $n/2$ para ver si se encuentran datos repetidos en dichos subarreglos.

Palabras Clave: Arreglos, Numeros Aleatorios, SubArreglos, Gráfica, C++.

1 Introducción

Para poder entender lo que se está haciendo en esta práctica, primero definamos un algoritmo.

Un algoritmo se define como el conjunto ordenado y finito de operaciones que permite hallar la solución de un problema. Esto es bastante importante, ya que básicamente la mayoría de las tareas que realiza una computadora se basan en algoritmos.

¿Por qué estudiarlos?

La principal razón para estudiarlos es para saber cual es mas eficiente, puesto que, aunque 2 algoritmos pueden hacer la misma tarea, no significa que a ambos les tome el mismo tiempo realizar una tarea, esto tiene un impacto aun más grande cuando las cantidades de datos que se manejan son demasiado grandes.

2 Conceptos Básicos

En esta practica graficaremos los puntos experimentales, independientemente si estos son los mejores o los peores casos.

Θ : Son los puntos experimentales, lo que nuestro algoritmo encuentra.

O : Es el peor caso, que se da cuando nuestro algoritmo recorre todas las posiciones de ambos sub arreglos.

Ω : es el mejor caso, que se da cuando nuestro algoritmo encuentra el numero repetido en el ultimo lugar de ambos sub arreglos.

Para la realización de esta practica hicimos uso de un algoritmo de búsqueda, el cual se fijará en el sub arreglo $A1[n/2]$ y verificara si hay algún numero repetido en el arreglo $A2[n/2]$, ambos sub arreglos salieron de un arreglo $A[n]$ el cual está lleno de números aleatorios de $0 - 3N$, el arreglo fue dividido en dos partes.

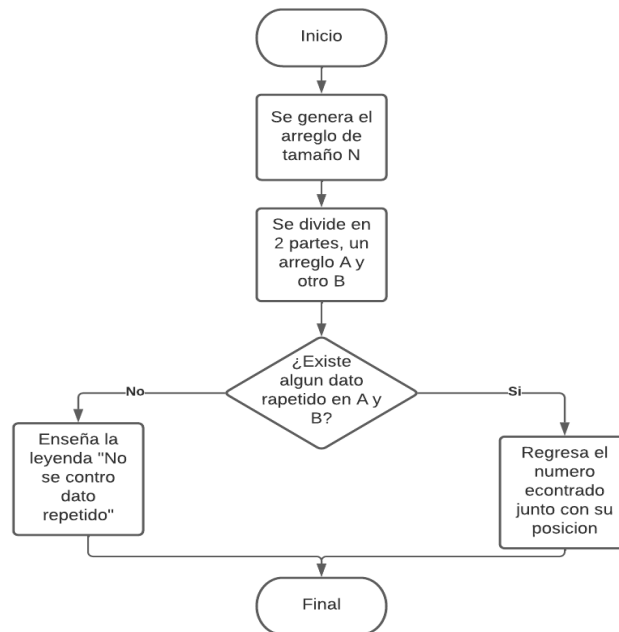


Figura 1. Diagrama Flujo.

3 Experimentación y Resultados

El programa va a buscar en arreglos de tamaño 10, este va a aumentar de 10 en 10 hasta llegar a 1000, por lo que en la gráfica contaremos con 100 registros para así observar el comportamiento del algoritmo. Nos basamos en el primer arreglo de 10 caracteres, podemos observar el comportamiento del algoritmo y por ende su resultado.

```
18 15 22 8 5 24 25 0 8 22
      Dato repetido: 22 Posiciones: 2,9
Tiempo: 104      Datos: 10
```

Figura 2. Arreglo de tamaño 10 con coincidencia.

```
21 1 29 16 5 14 23 19 2 19
No hay dato repetido
```

Figura 3. Arreglo de tamaño 10 sin coincidencia.

Al momento de graficar los puntos experimentales encontramos la siguiente imagen incluyendo las funciones por la cuales es acotada $[G(n)$ y $F(n)]$.

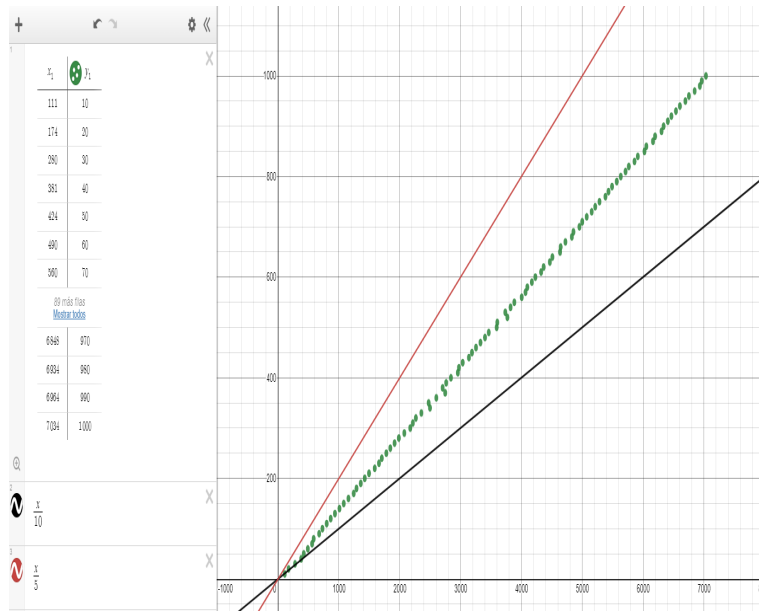


Figura 4. Gráfica con $G(n) = x/5$ y $F(n)=x/10$.

4 Conclusiones

Leon Tejada:

El análisis de algoritmos tiene un papel muy importante, ya que podríamos encontrar un método mucho más eficiente para resolver algún problema, también nos ayuda a ver si nuestra puede mejorar o no. Aquí es en donde más tienen que ver las matemáticas, o donde más las vemos aplicadas, debido a que incluso se pueden ver series al momento de estar analizando algún código.

