1. Calcule la eficiencia empírica, siguiendo las indicaciones de la sección 3. Defina adecuadamente los tamaños de entrada de forma tal que se generen al menos 25 datos. Incluya en la memoria tablas diferentes para los algoritmos de distinto orden de eficiencia (una con los algoritmos de orden O(n2), otra con los O(n log n), otra con O(n3) y otra con O(2n)).

2. Con cada una de las tablas anteriores, genere un gráfico comparando los tiempos de los algoritmos. Indique claramente el significado de cada serie. Para los algoritmos que realizan la misma tarea (los de ordenación), incluya también una gráfica con todos ellos, para poder apreciar las diferencias en rendimiento de algoritmos con diferente orden de eficiencia.

3. Calcule también la eficiencia hibrida de todos los algoritmos, siguiendo las pautas indicadas en la sección 4. Pruebe también con otros ajustes que no se correspondan con la eficiencia teórica (ajuste lineal, cuadrático, etc) y compruebe la variación en la calidad del ajuste.

4. Otro aspecto interesante a analizar mediante este tipo de estudio es la variación de la eficiencia empírica en función de parámetros externos tales como: las opciones de compilación utilizada (con/sin optimización), el ordenador donde se realizan las pruebas, el sistema operativo, etc. Sugiera algún estudio de este tipo, consulte con el profesor de prácticas y llévelo a cabo.

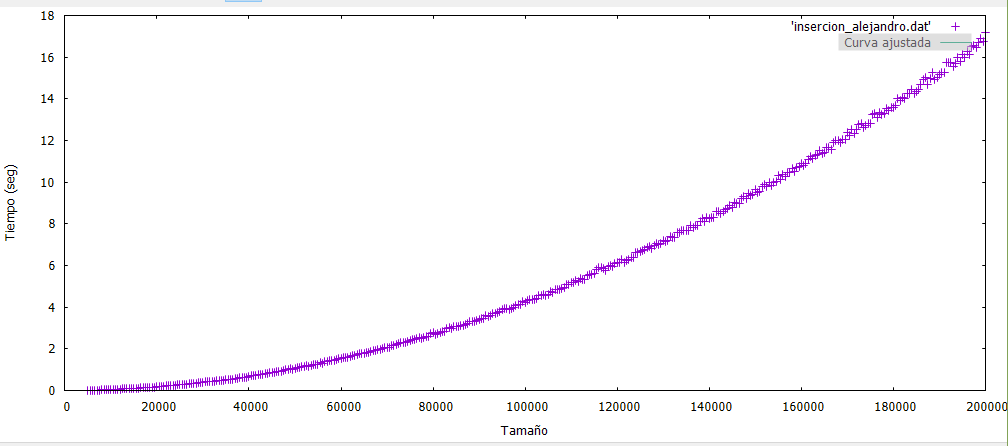
5. Escriba una memoria con todas las tareas realizadas, incluyendo todos los detalles relevantes referidos a los cálculos de eficiencia empírica e híbrida. El informe debe entregarse en formato pdf.

Algoritmo inserción

A continuación, calcularemos la eficiencia híbrida del algoritmo de inserción. En primer lugar, ejecutamos el algoritmo con un valor de inicio de 5000, un valor de fin de 200000 y un incremento de 500. Un resumen de estos valores:

|  |  |
| --- | --- |
| Tamaño de entrada | Tiempo de ejecución |
| 6000 | 0.015625 |
| 8000 | 0.03125 |
| 9500 | 0.046875 |
| 107500 | 4.85938 |
| 108000 | 4.90625 |
| 108500 | 4.95312 |
| 199000 | 16.8906 |
| 199500 | 16.7344 |
| 200000 | 17.1719 |

Si representamos todos los valores en una gráfica tenemos:



Como podemos apreciar al tener muchos valores la gráfica pertenece a O(n2)

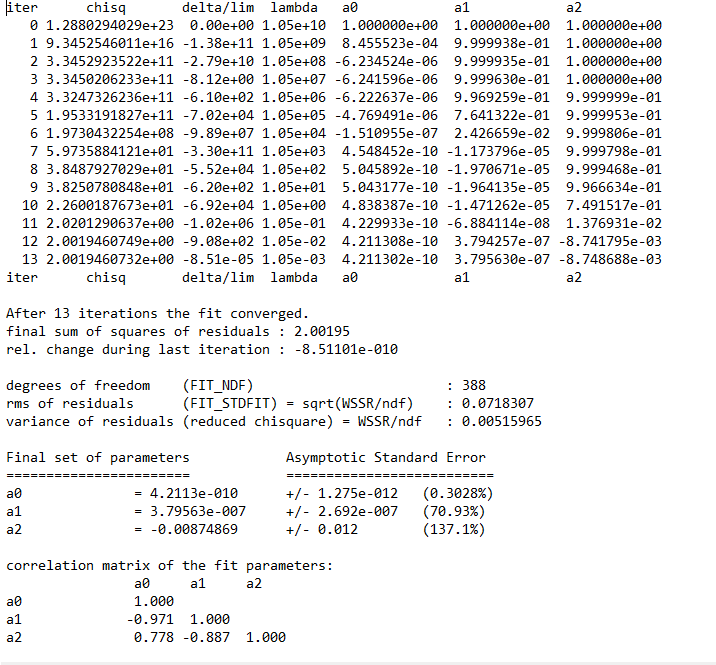
La función a ajustar a los puntos obtenidos en el cálculo de la eficiencia empírica será: T(n) = a0\*x\*x+a1\*x+a2 . Una vez calculada la regresión en gnuplot tenemos los valores de las constantes:

a0 = 4.2113e-010

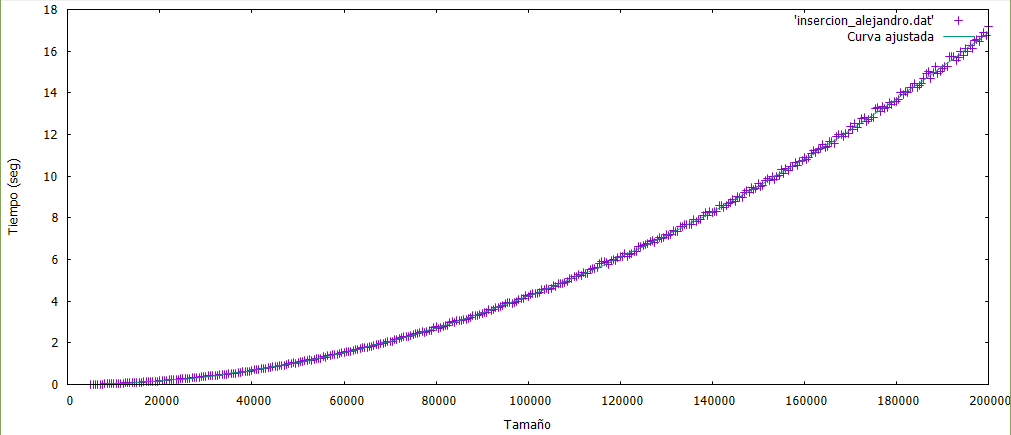
a1 = 3.79563e-007

a2 = -0.00874869

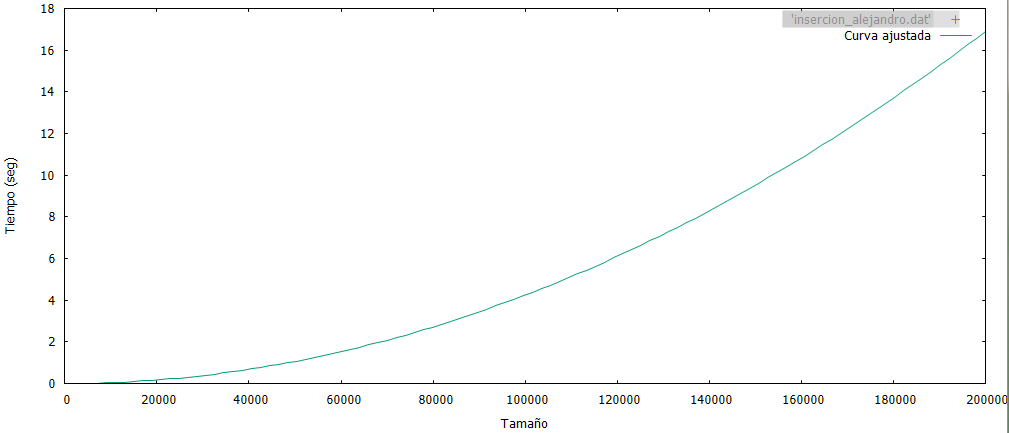
El resultado de la regresión ha sido:



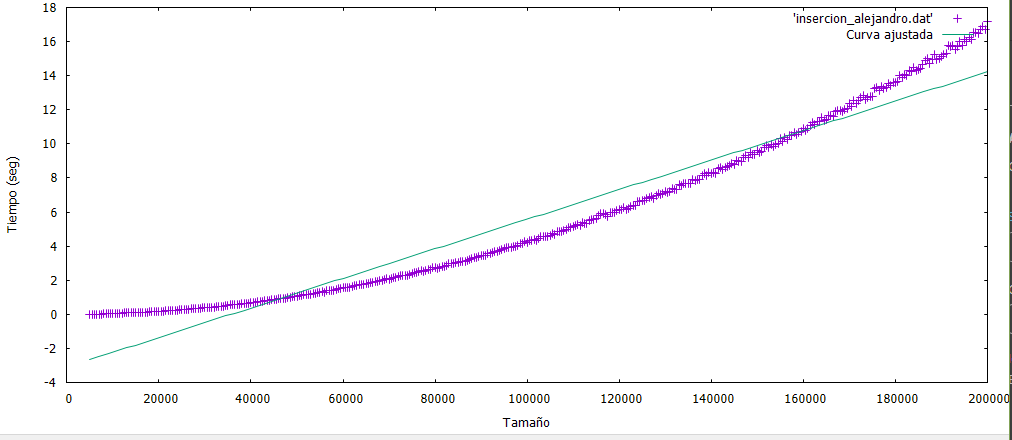
La gráfica de los puntos y el ajuste es:



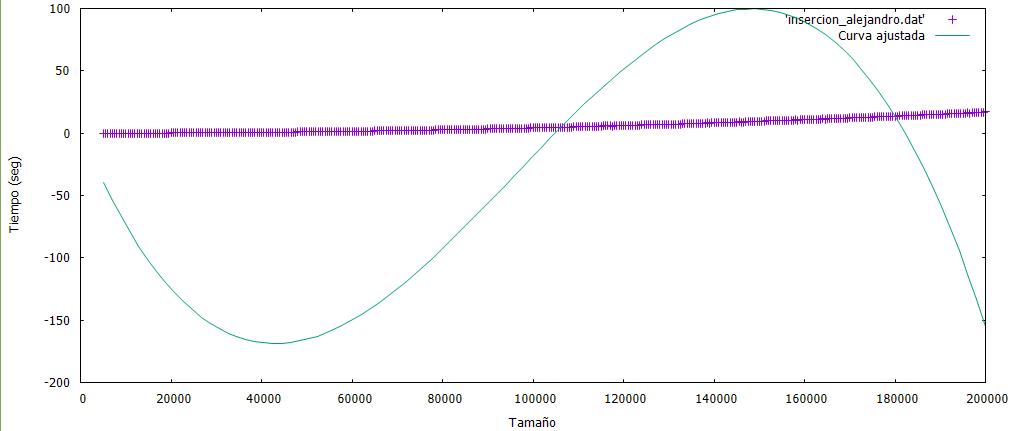
Si quitamos los puntos de los valores de las ejecuciones obtenemos:



Podemos apreciar que la función se ajusta muy bien a la gráfica de puntos. En cambio, si ajustamos otras funciones de distintos órdenes de eficiencia vemos que el ajuste es pésimo ya que no se corresponde con la gráfica de puntos:

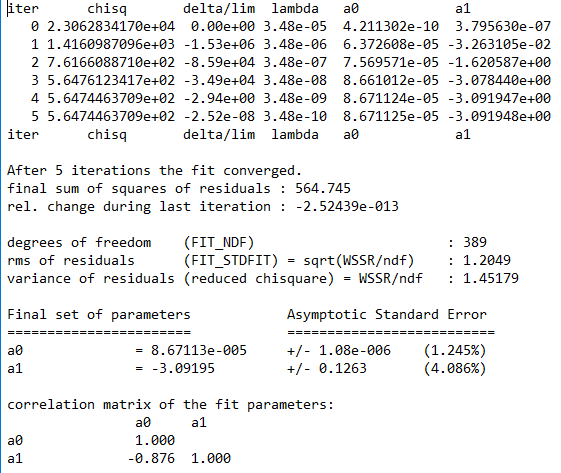


Ajuste lineal O(n) con una función T(n) = 8.67113e-005\*x-3.09195

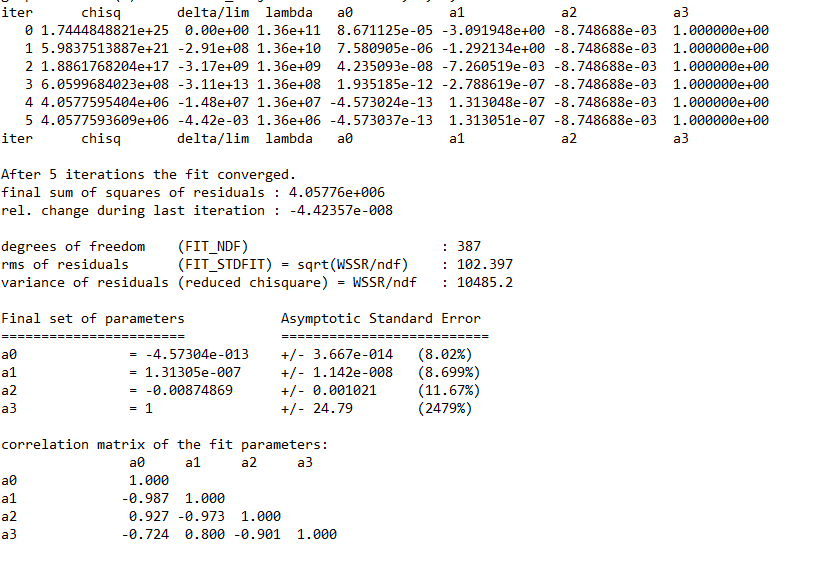


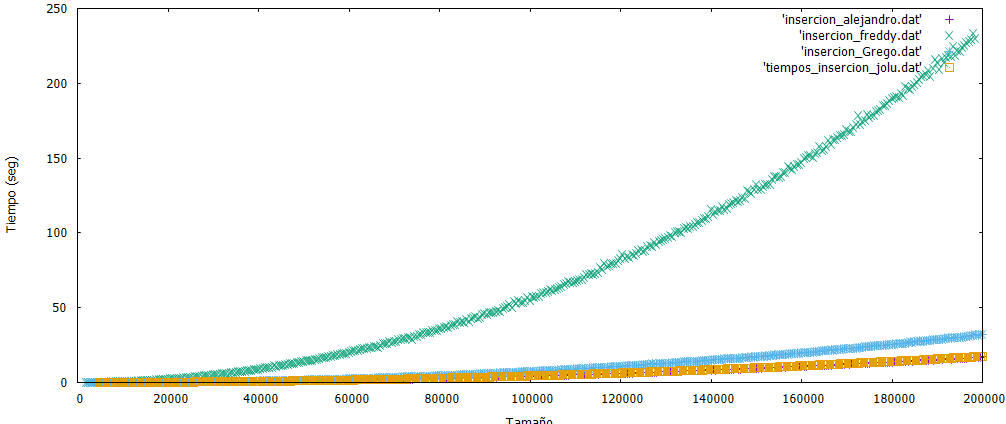
Ajuste cúbico O(n3) con una función T(n) = -4.57304e-013\*x\*x\*x+1.31305e-007 \*x\*x-0.00874869\*x+1

El resultado de la regresión del ajuste lineal ha sido:



El resultado de la regresión del ajuste cúbico ha sido:



Si juntamos todas las gráficas de los ordenadores que han ejecutado este algoritmo tenemos:

En ella se pueden apreciar que los tiempos de ejecución son distintos según el ordenador que los ha ejecutado. Esto se debe a las diferencias de hardware de cada ordenador y también a la optimización a la hora de compilar el código. Los tiempos de ejecución de los archivos ‘insercion\_alejandro.dat’ y ‘tiempos\_insercion\_jolu.dat’ han sido compilados con la opción -O2