Índice

1. Análisis problema
2. Soluciones
   1. Fuerza Bruta
   2. Divide y vencerás
3. Eficiencia empírica
4. Eficiencia híbrida
5. Conclusiones

# Análisis del problema

Para esta práctica se nos planteaba el problema de encontrar las diferencias entre dos vectores los están ordenados según unas preferencias.

Para ello suponemos un vector con un tamaño **N** que está ordenado de menor a mayor y otro vector con un mismo tamaño **N** y con los mismos elementos que el primero colocados en posiciones aleatorias.

Para contar las inversiones de un vector respecto de otro compararemos todas las posiciones de un vector con las que equivaldrían en el otro vector y si el valor es distinto aumentaremos un contador de inversiones.

Una vez hayamos recorrido ambos vectores tendremos el total de inversiones que presentan uno respecto del otro.

# Soluciones

Al intentar resolver este problema nos hemos encontrado con varios inconvenientes.

El primero se trata del orden de eficiencia que tendría nuestro algoritmo comparados pues para comparar la misma posición de dos vectores sólo nos haría falta un bucle iterativo, lo que haría que la eficiencia en el peor de los casos fuera lineal.

El segundo, siendo consecuencia de este primero es, ¿Para qué vamos a aplicar la técnica de Divide y Vencerás si nuestro algoritmo ya tiene una eficiencia relativamente pequeña?

Para solucionar este último hemos supuesto otro planteamiento del problema, supondremos que las preferencias de un vector será que los primeros elementos siempre sean menores que los que le siguen (un vector ordenado) lo cual hace que ahora solo tengamos un vector de elementos y que comparemos estos mismos para comprobar las inversiones respecto a las preferencias ideales.

En este caso tenemos un vector en el que comparamos todos los elementos NxN veces, lo que sí nos permitiría aplicar Divide y Vencerás para hacer un algoritmo más rápido.

## Fuerza bruta

En nuestra primer acercamiento al problema, utilizamos un método por fuerza bruta, el cual consta de dos bucles for, uno anidado dentro de otro. Lo que hace este algoritmo es tomar el vector que se quiere comparar y comprobar el elemento que está en la posición i con cada uno de los elementos que le siguen si es mayor, si es así entonces hay una inversión y se actualiza el contador de inversiones, y asi se hace con cada uno de los elementos del vector.

Como se puede apreciar el algoritmo al ser bucles anidados su eficiencia es O(n²) pues el bucle interno es de eficiencia n, y el externo igual así la eficiencia de este algoritmo sería nxn, O(n²)

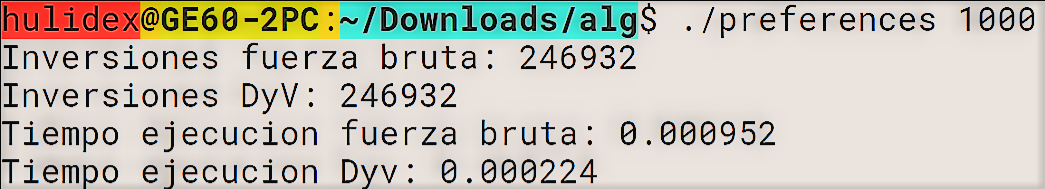
## Divide y vencerás

Para el planteamiento del problema, contamos con un solo vector, suponemos que el vector ordenado son los índices del vector, sólo nos fijamos entonces en el contenido del vector que es lo que va a tener las inversiones. Para realizar el conteo tendremos que dividirlo y contar el número de inversiones recursivamente será una opción que nos permita obtener un orden de eficiencia menor al cuadrático.

Para resolverlo mediante la técnica de divide y vencerás hemos optado por usar el algoritmo ***mergesort.***

Usando este algoritmo conseguimos ir separando el vector y contando las inversiones en vectores más pequeños hasta que no podamos separar más el vector. Una vez llegados a este punto el vector se mezcla y se cuentan las inversiones unidos estos dos vectores.

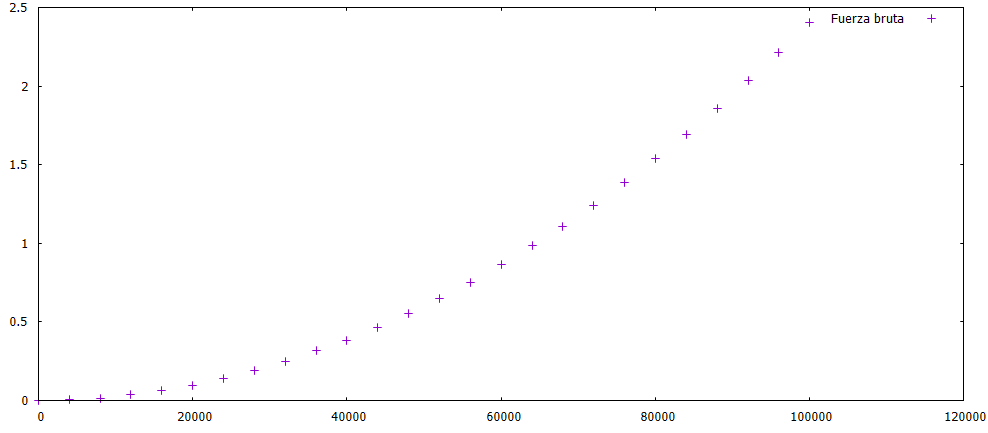
Al llegar al punto en el que todo el vector se haya unido habremos obtenido el total de inversiones el cual deberá coincidir con las inversiones totales que obtenemos usando el algoritmo de fuerza bruta.



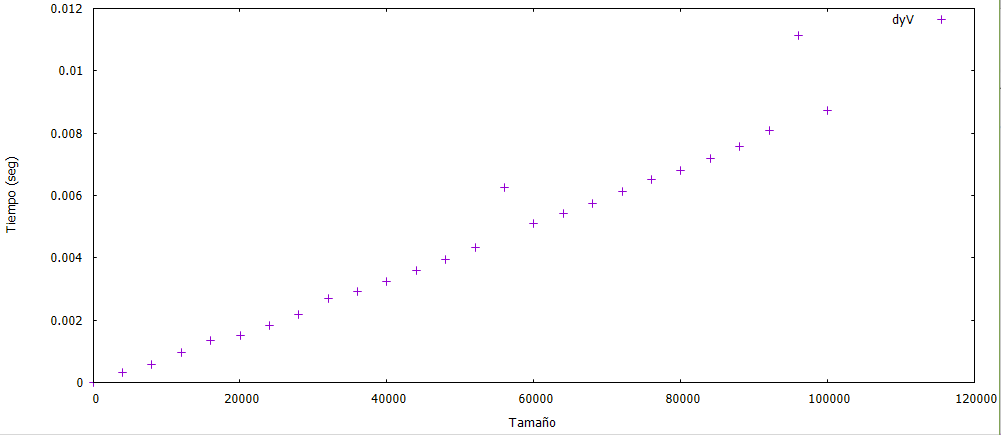
# Eficiencia empírica

Para calcular la eficiencia empírica hemos ejecutado los algoritmos con valores desde 1 hasta 100000 con un incremento de 4000 y se han compilado con una optimización de -O2

A continuación, se muestran las gráficas de ambos algoritmos:



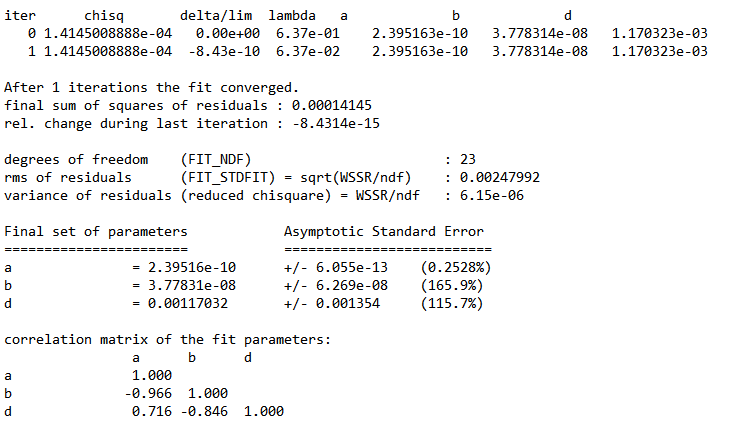
**Gráfica del algoritmo de fuerza bruta**



**Gráfica del algoritmo de divide y vencerás**

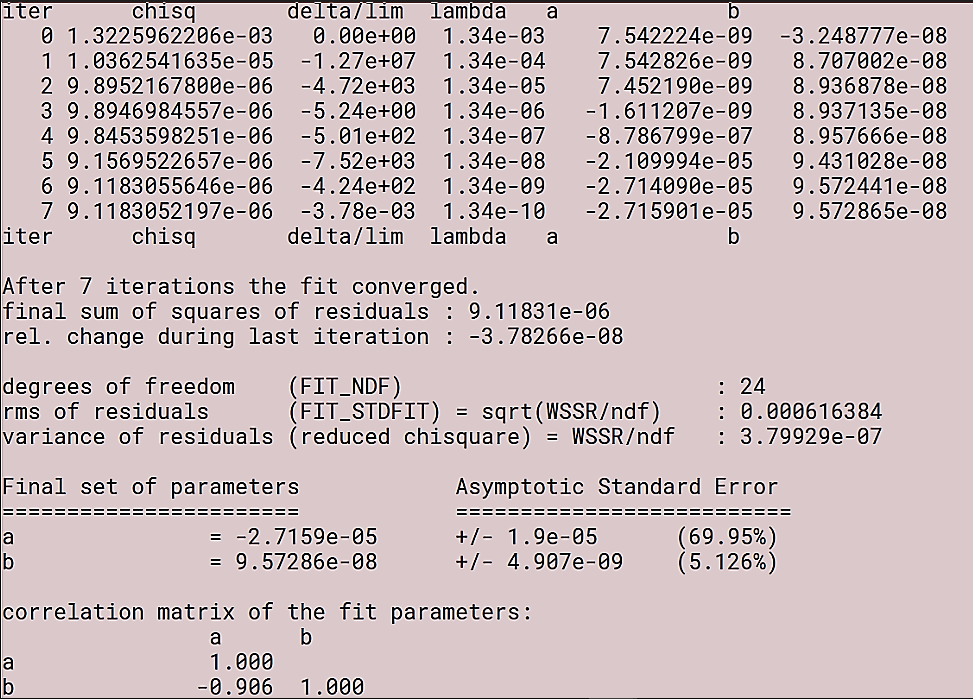
# Eficiencia híbrida

En el algoritmo de fuerza bruta ajustamos a la función F(x) = a0\*x^2+a1\*x+a2 y calculamos los coeficientes a0, a1 y a2

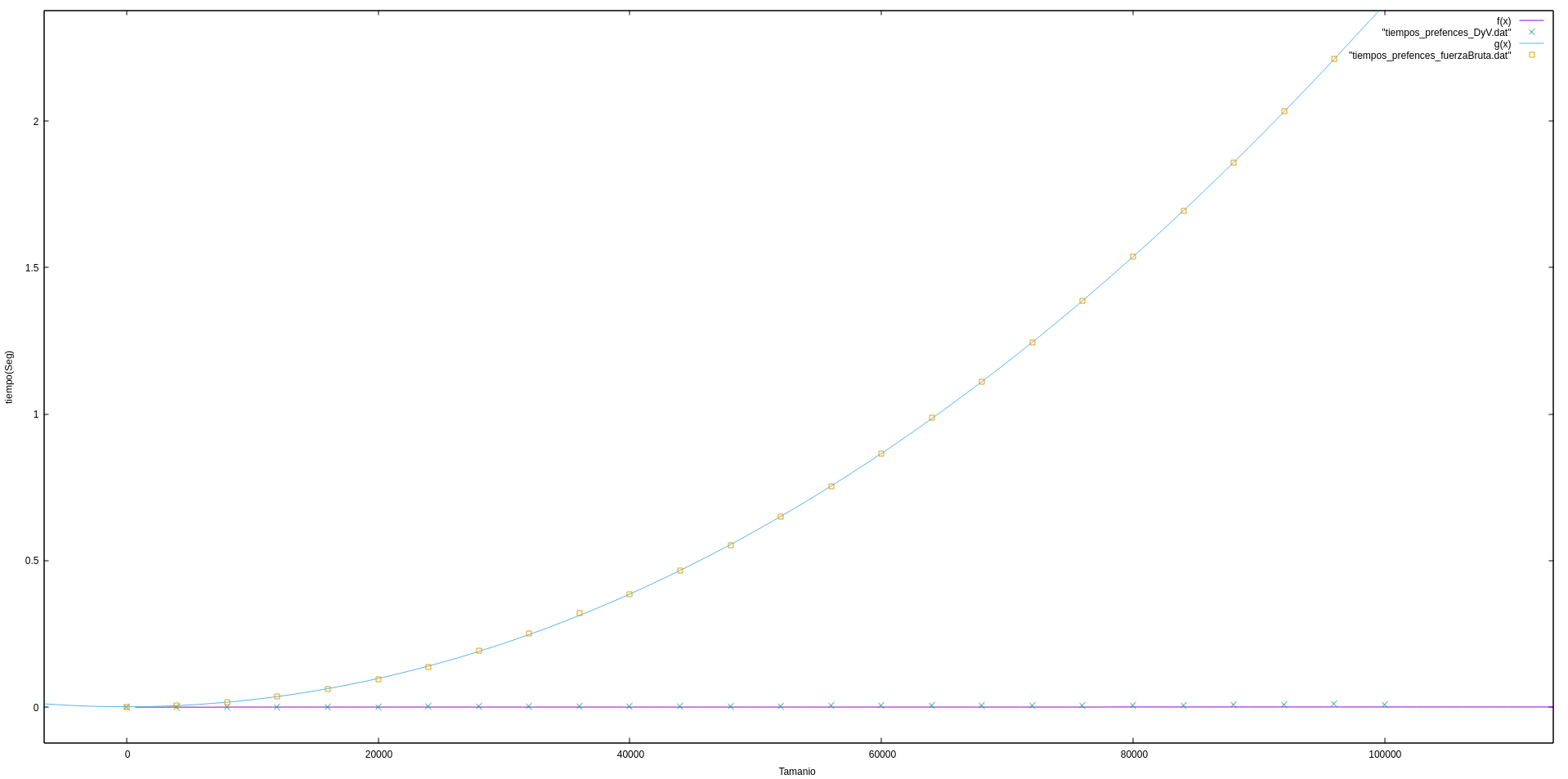


Como podemos ver el coeficiente de regresión de n^2 es cercano a 0 por lo que el ajuste realizado es muy bueno.

En el caso del algoritmo divide y vencerás la función a ajustar es G(x) = a\*x\*(log10(b\*x)/log10(2))+c



Obtenemos los valores de los coeficientes a y b. Como se puede ver en la siguiente gráfica el ajuste realizado es muy bueno.



# Conclusiones

Al aplicar Divide y vencerás conseguimos mejorar la eficiencia de nuestro programa, ya que en lugar de tener un orden cuadrático de eficiencia lo rebajamos a un nlog(n).

Dividimos el problema en subproblemas más pequeños en los que se usa la mitad del tiempo de cómputo de lo que se hacía si no lo dividimos y una vez que hemos descompuesto en subproblemas y tenemos soluciones parciales, analizamos las soluciones parciales para conseguir el resultado final, recomponiendo las soluciones para llegar a una única solución. La solución final del problema.