

ESTRUCTURAS DE DATOS

PRACTICA 1

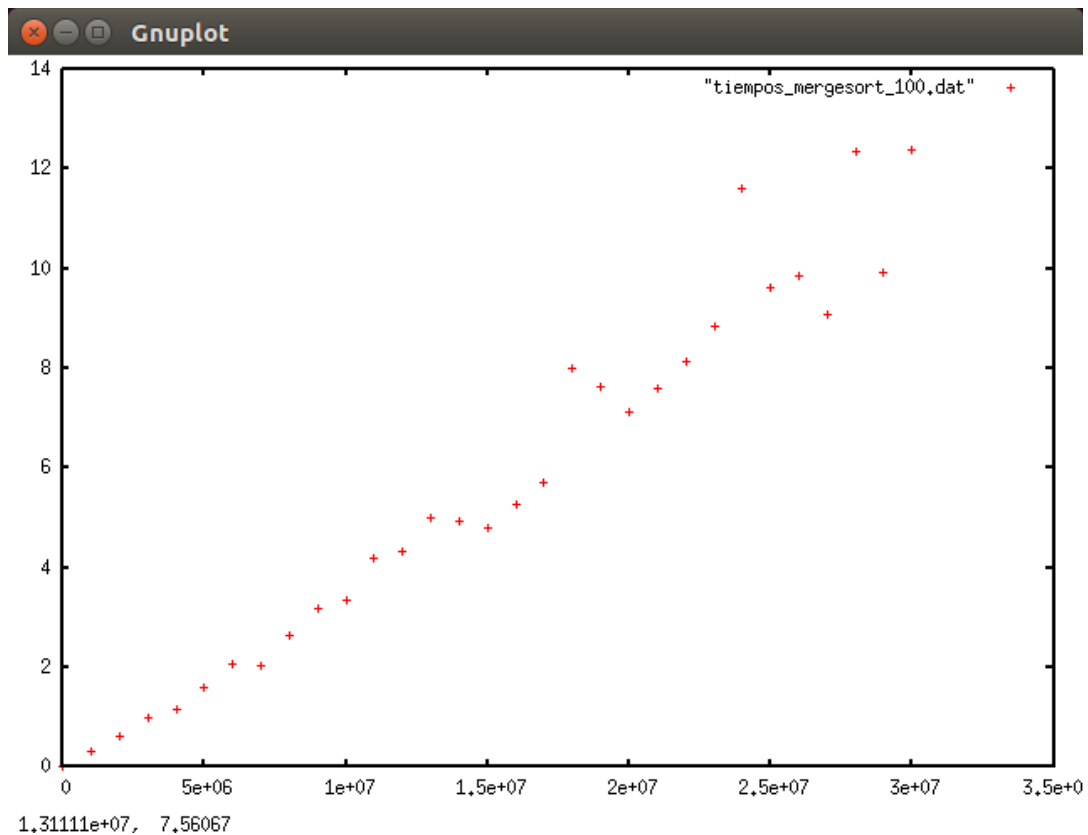
8) Ordenación por mezcla:

Análisis empírico de la eficiencia:

```
#!/bin/csh
@ inicio = 100
@ fin = 30000100
@ incremento = 1000000
set ejecutable = mergesort
set salida = tiempos_mergesort.dat

@ i = $inicio
echo > $salida
while ( $i <= $fin )
    echo Ejecución tam = $i
    echo `./{$ejecutable} $i` >> $salida
    @ i += $incremento
end
```

Representación de los tiempos obtenidos tras la ejecución del script anterior:



Captura de datos realizada para un valor UMBRAL_MS = 100.

Ajuste regresión:

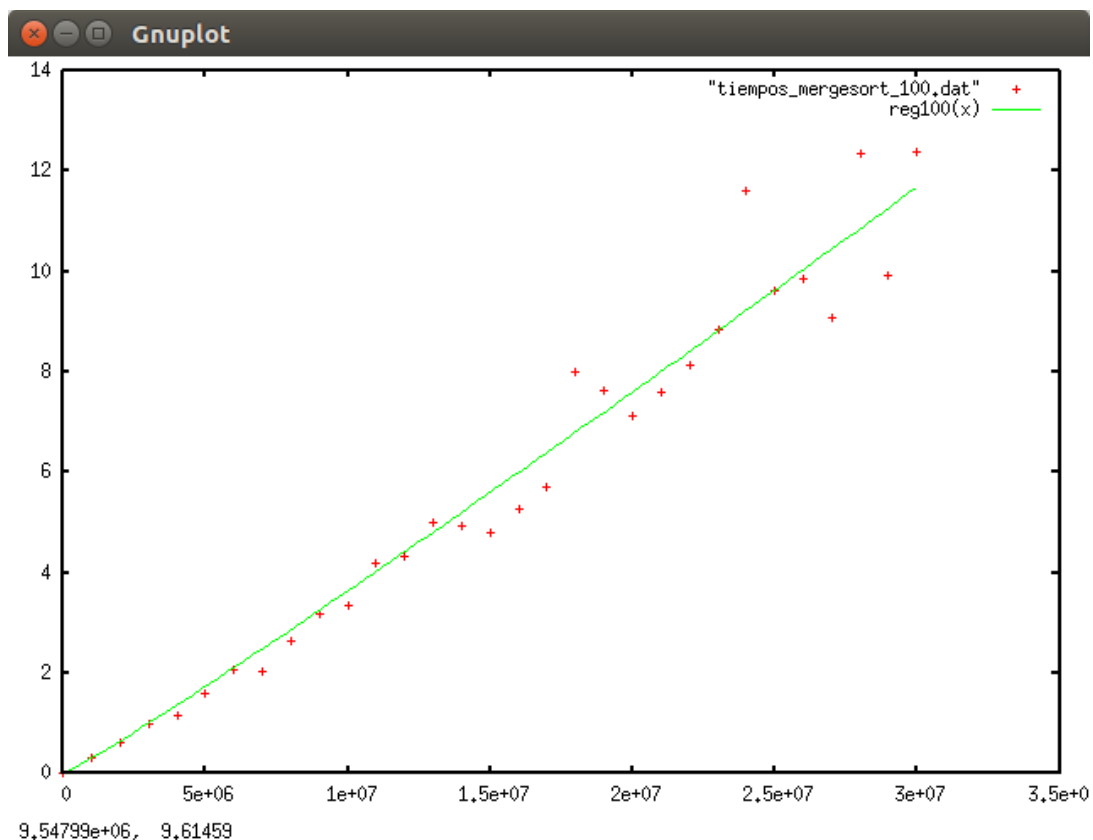
```
After 5 iterations the fit converged.
final sum of squares of residuals : 16.4049
rel. change during last iteration : -2.16564e-16

degrees of freedom    (FIT_NDF)           : 30
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)   : 0.73948
variance of residuals (reduced chisquare) = WSSR/ndf  : 0.54683

Final set of parameters      Asymptotic Standard Error
=====
a          = 2.2611e-08      +/- 4.499e-10    (1.99%)

correlation matrix of the fit parameters:

          a
a         1.000
```



Captura de datos realizada para un valor UMBRAL_MS = 100.

Estudio acerca de la variable UMBRAL_MS.

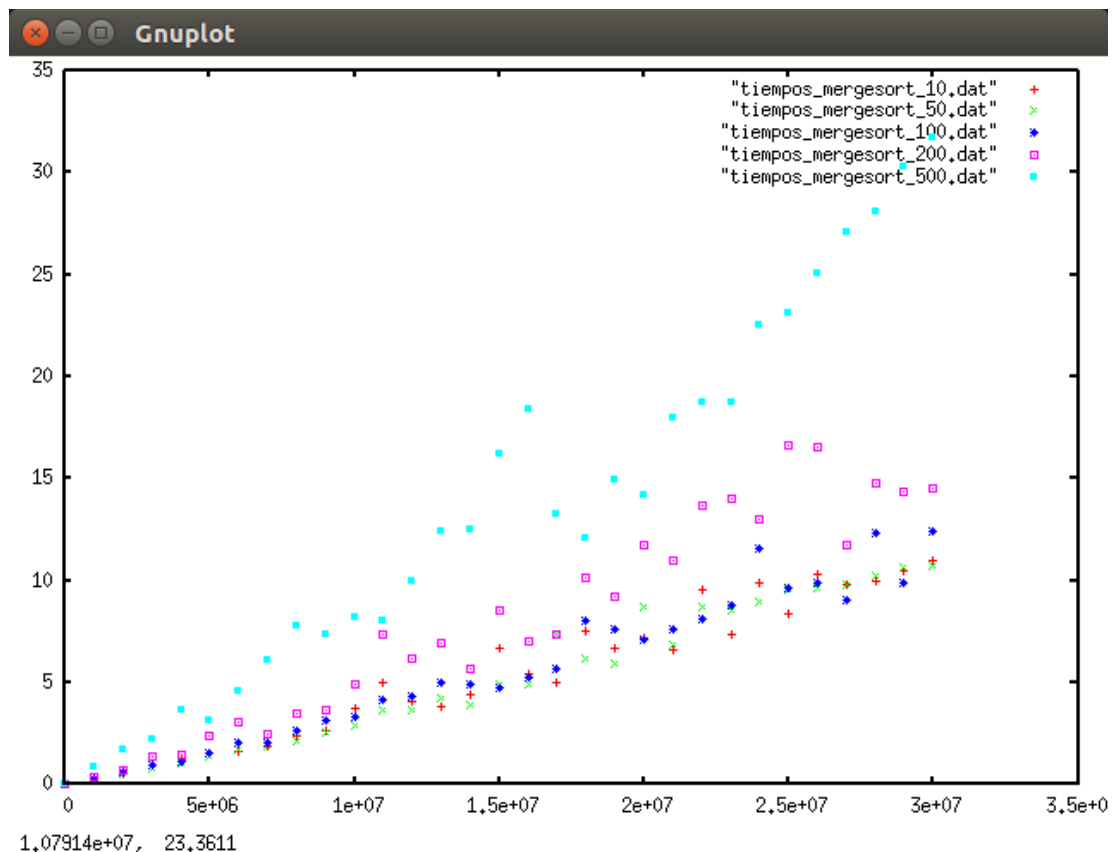
A simple vista podemos detectar que la variable UMBRAL_MS controla si el algoritmo ordenará el vector mediante el método de inserción (cuya eficiencia es $O(n^2)$) o si lo hará con el método mezcla (cuya eficiencia es $O(n \log(n))$).

El mejor o peor resultado en los tiempos de ejecución dependerá del número de veces que ejecutemos el algoritmo mezcla, es decir, el tiempo de ejecución será menor cuantas más veces se ejecute el algoritmo mezcla frente al algoritmo de inserción; o dicho de otro modo, cuanto mayor sea UMBRAL_MS, más veces ejecutaremos el algoritmo de inserción y por consiguiente mayor será el tiempo de ejecución.

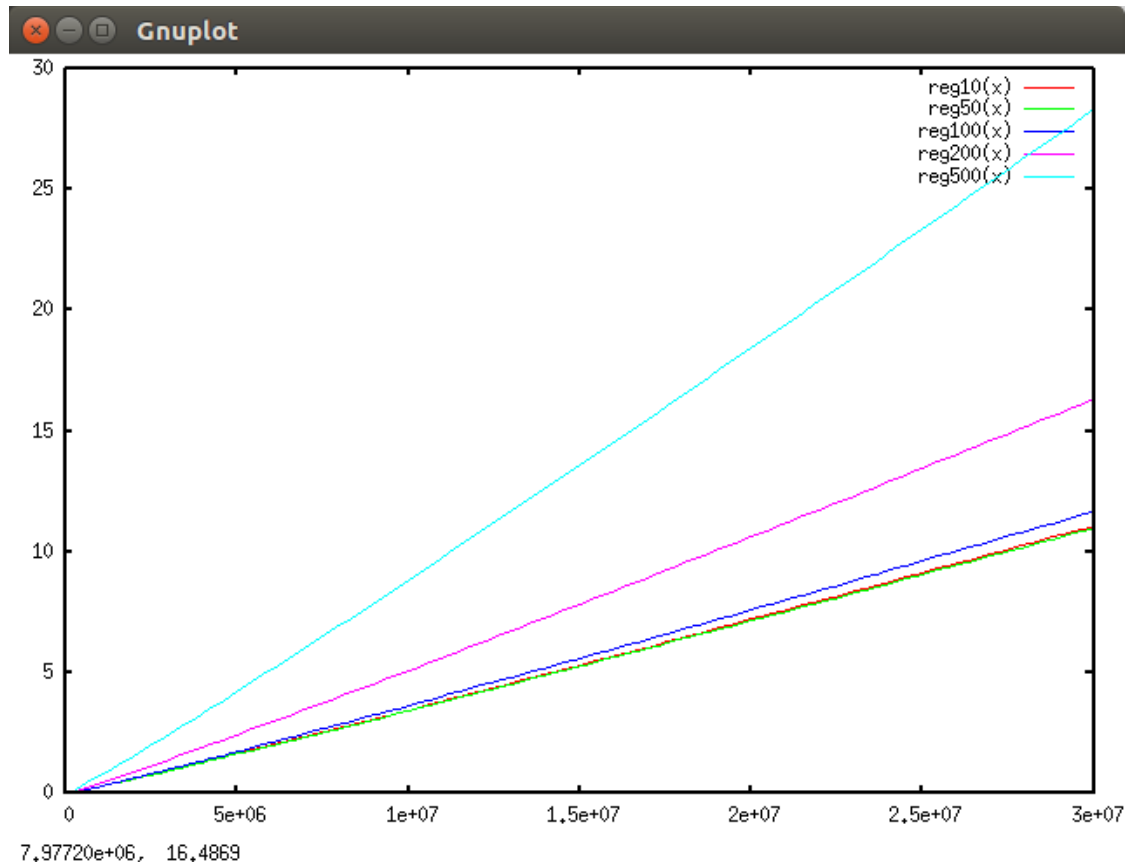
De lo anterior sabemos que hay que buscar un valor bajo para UMBRAL_MS y dicho valor debe de ser lo suficientemente bajo como para que sea indiferente ejecutar el algoritmo mediante un método u otro para un tamaño de vector determinado, con las gráficas ayudaremos a deducir dicho valor.

El estudio consiste en almacenar los tiempos obtenidos al ejecutar el script para los siguientes valores para UMBRAL_MS (en el fichero mergesort.cpp): 10, 50, 100, 200 y 500, y así ver cómo afecta el valor de UMBRAL_MS al tiempo de ejecución, la idea es ver cuando no podemos mejorar más el tiempo de ejecución por más que bajemos el valor de UMBRAL_MS.

A continuación mostraré una gráfica correspondiente a los valores de de los tiempos obtenidos con los valores para UMBRAL_MS citados anteriormente:



Una gráfica con los ajustes de regresión nos ayudará a ver los resultados con más claridad:



Efectivamente, podemos comprobar que cuanto mayor sea el valor de la variable UMBRAL_MS, mayor será el tiempo de ejecución.

La conclusión que podemos sacar del análisis es que la mejora en tiempo de ejecución del algoritmo con UMBRAL_MS = 10 frente al algoritmo con UMBRAL_MS = 50 es inapreciable incluso para vectores de tamaño 30.000.000, luego el valor ideal en mi equipo para UMBRAL_MS es cualquier valor comprendido en el intervalo [10,50].