

# Estudio empirico e hibrido de la eficiencia de algoritmo divide y venceras vs algoritmo lineal

## Enunciado del ejercicio

Implementar un algoritmo sencillo para calcular el valor máximo de un vector y otro basado en divide y vencerás. Realizar un estudio empírico e híbrido de su eficiencia.

## Toma de datos

Junta a este pdf, incluyo los codigos que he usado para la realización del estudio, asi como los datos obtenidos en los experimentos.

Una vez realice todo el codigo pertienen, me puse a buscar un umbral empirico adecuado para el algoritmo divide y venceras. Puse diferentes umbrales desde 1.000 hasta 1.000.000 realizando incrementos de 100.000. Pero no obtuve ningun umbral mas adecuado que otro, por lo tanto elegi de todos ellos el que menor tiempo tardo, que fue el 101.000. Aunque ninguno de ellos me dio un tiempo considerablemente mejor que los otros.

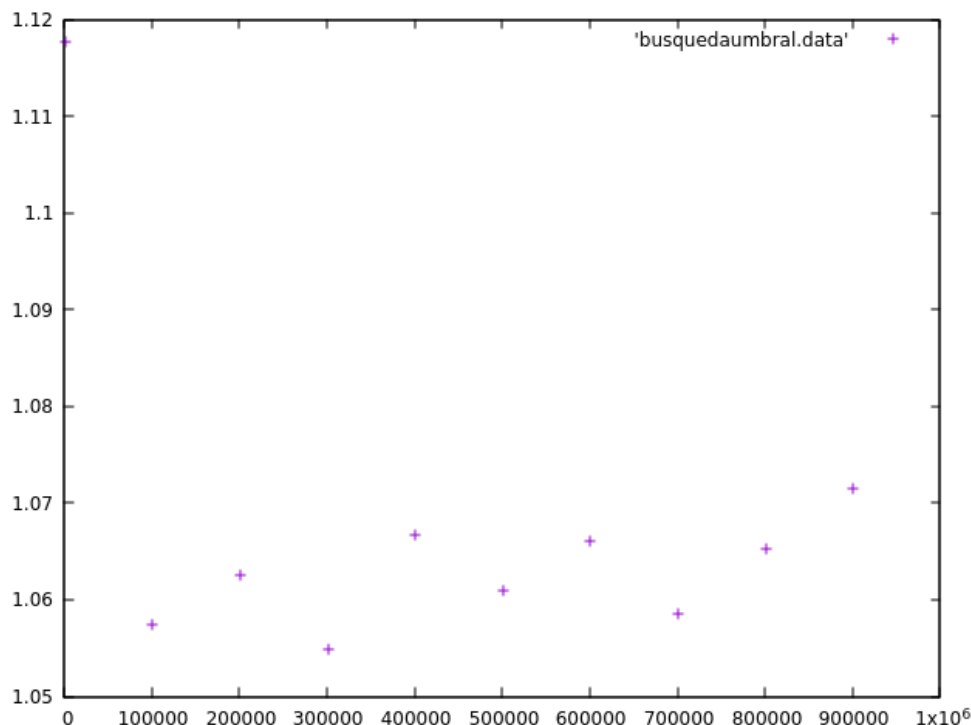
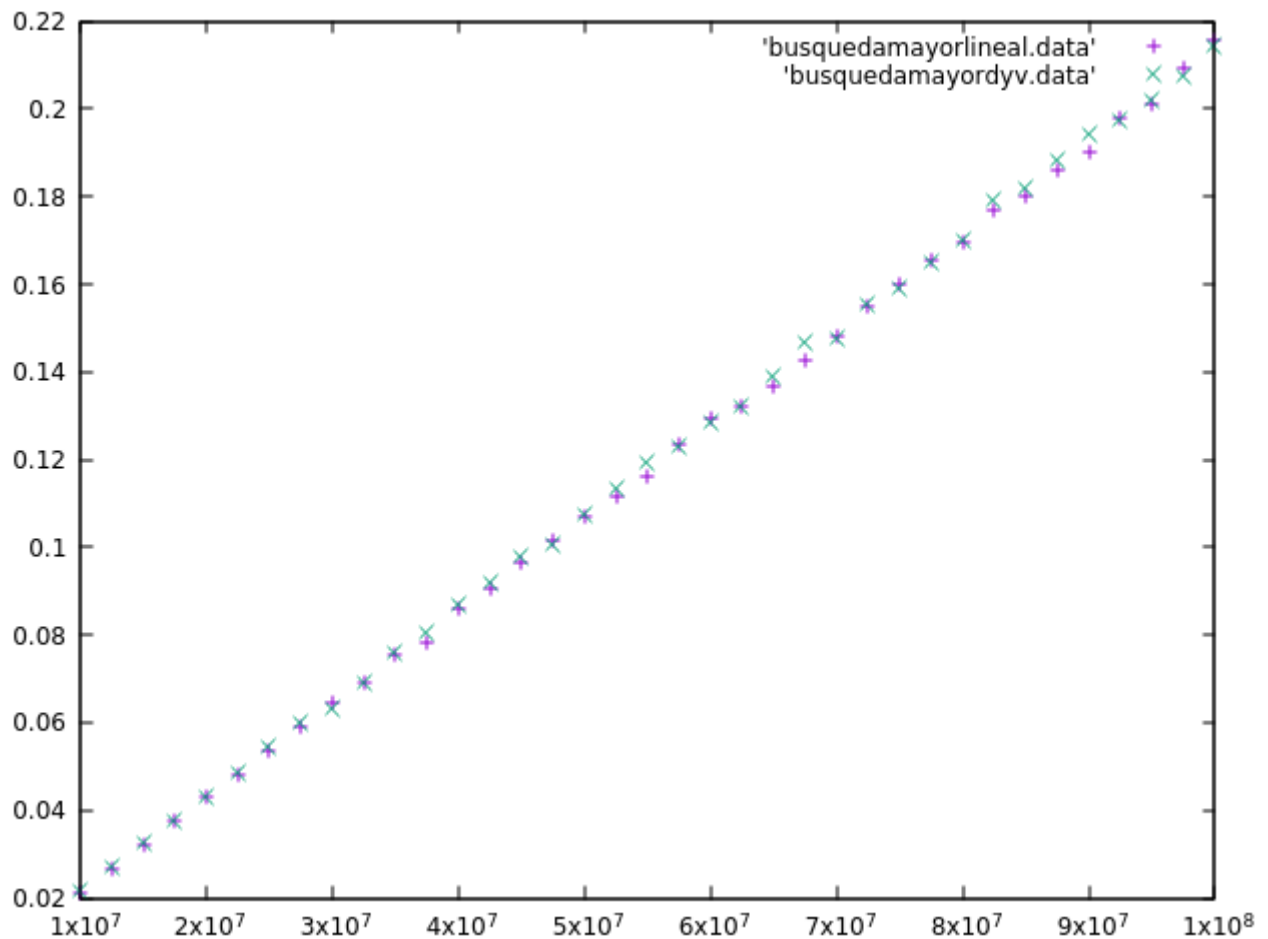


Ilustración 1: Grafica de los umbrales probados

Acto seguido me dispuse a obtener los tiempos para diferentes tamaños de problema tanto para el algoritmo de búsqueda lineal como el que usa divide y venceras (adjunto los datos como fichero de texto junto a este pdf). Y obtuvimos que con los dos obteniamos tiempos muy parecidos empíricamente.

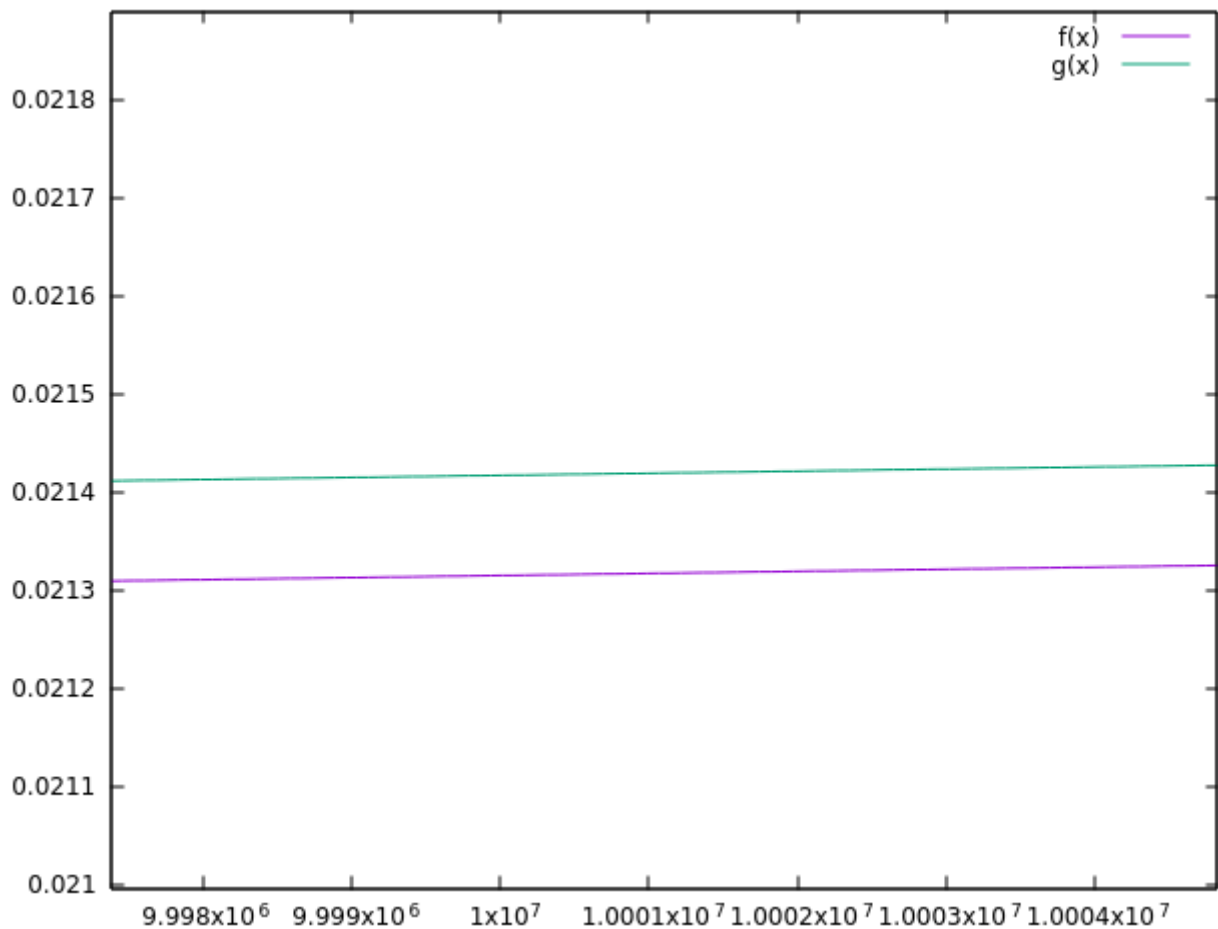


*Ilustración 2: Datos obtenidos de los dos algoritmos*

Al tener los dos las graficas tan parecidas me dispuse a comprobar su eficiencia hibrida, para comprobar cual de los dos tenia una mayor pendiente y asi poder saber cual de los dos es peor.

Hice el ajuste con el gnuplot y obtuve que la pendiente para el algoritmo que no usaba divide y venceras tenia una constante de  $2.13145 \times 10^{-9}$  y el que si usaba divide y venceras tenia una constante de  $2.14166 \times 10^{-9}$ .

En el grafico he puesto un intervalo pequeño, para que asi si se pueda apreciar mejor la diferencia entre las distintas pendientes. En  $g(x)$  tenemos al algoritmo divide y venceras y en  $f(x)$  tenemos al algoritmo lineal normal.



*Ilustración 3: Eficiencia hibrida*

Por lo tanto aunque los dos tiene tiene pendientes muy parecidas, que el algoritmo divide y venceras tenga mas pendiente, se debe a que tiene que hacer mas apilamientos de subrutinas y mas llamada al sistema, por lo que le da un peor resultado.