

Fibonacci

Explicación del Algoritmo

Se trata de un algoritmo, basado en la ecuación:

$$f_n = f_{n-1} + f_{n-2}$$

Que nos permite calcular de forma recursiva la Sucesión de Fibonacci, se trata de un algoritmo de orden $O(\left(\frac{1+\sqrt{5}}{2}\right)^n)$. Esto produce una recurrencia que aumenta bastante en tiempo de ejecución conforme aumentan el tamaño de los números (n)

Tiempos

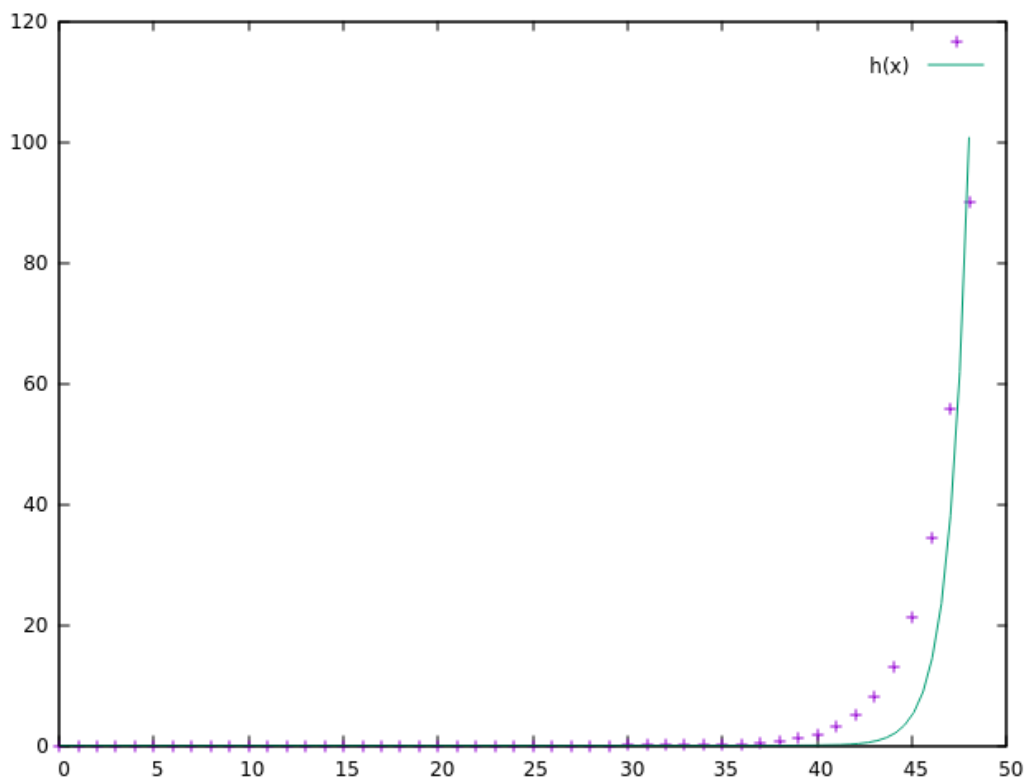
Para la medición de los tiempos, se ha usado un método que mide el tiempo antes y después de cada una de las ejecuciones con diferentes tamaños para el 'n'.

Para el cálculo de grafica han sido tomadas una serie de medidas de tiempo, concretamente unas medidas para n entre 0 y 50.

Para esto se ha usado una función $h(x)=a_0*1.618$ obteniendo un error de:

a0 = 8.86664e-20 +/- 4.386e-21 (4.947%)

Representación Gráfica



Como se puede observar en la grafica, la adaptación se mejora conforme el tamaño crece, pero no es tan buena para números relativamente bajos.

Explicación del Algoritmo

Se trata de un algoritmo de ordenación basado en la ecuación:

$$T(n) = T(1) + T(n-1) + c * n$$

Permite realizar de forma recursiva la ordenación de un vector, por medio de un algoritmo de orden $O(n \log n)$. Esto produce una recurrencia que aumenta pero de una forma bastante menos agresiva si la comparamos con otros algoritmos.

Tiempos

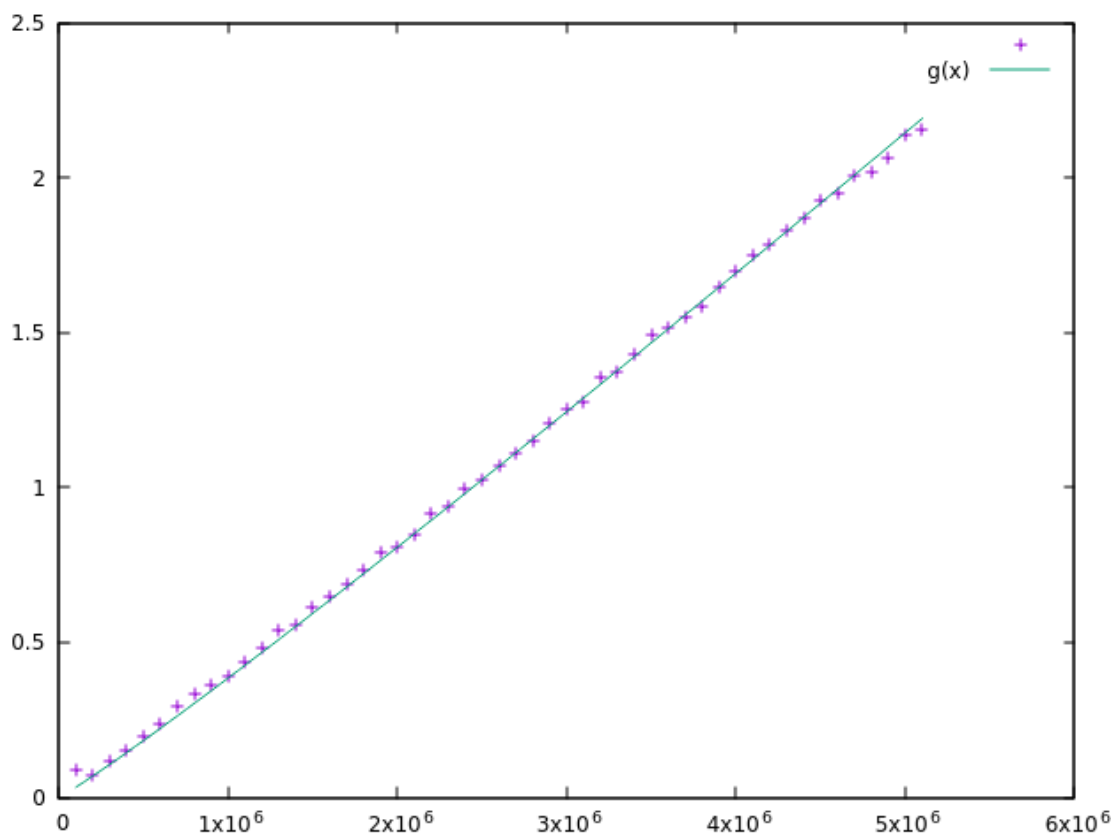
Para la medición de los tiempos, se ha usado un método que mide el tiempo antes y después de cada una de las ejecuciones con diferentes tamaños para el 'n'.

Para el cálculo de grafica han sido tomadas una serie de medidas de tiempo, concretamente unas medidas para n entre 100000 y 5100000.

Para esto se ha usado una función $g(x)=a0*(n \log n)$. obteniendo un error bastante bajo de:

$$a0 = 2.78128e-08 \quad +/- \quad 5.689e-11 \quad (0.2046\%)$$

Representación Gráfica



Como se puede observar en la gráfica, la función se adapta bastante bien a la crecida de los datos pero con algunas pequeñas variaciones.

Explicación del Algoritmo

Se trata de un algoritmo de ordenación que nos permite realizar de forma recursiva la ordenación de un vector, usa un algoritmo de orden $O(n^2)$. Que produce una recurrencia que va aumentando bastante, si no tanto como en algunas de un orden superior, si bastante conforme el valor de 'n' crece.

Tiempos

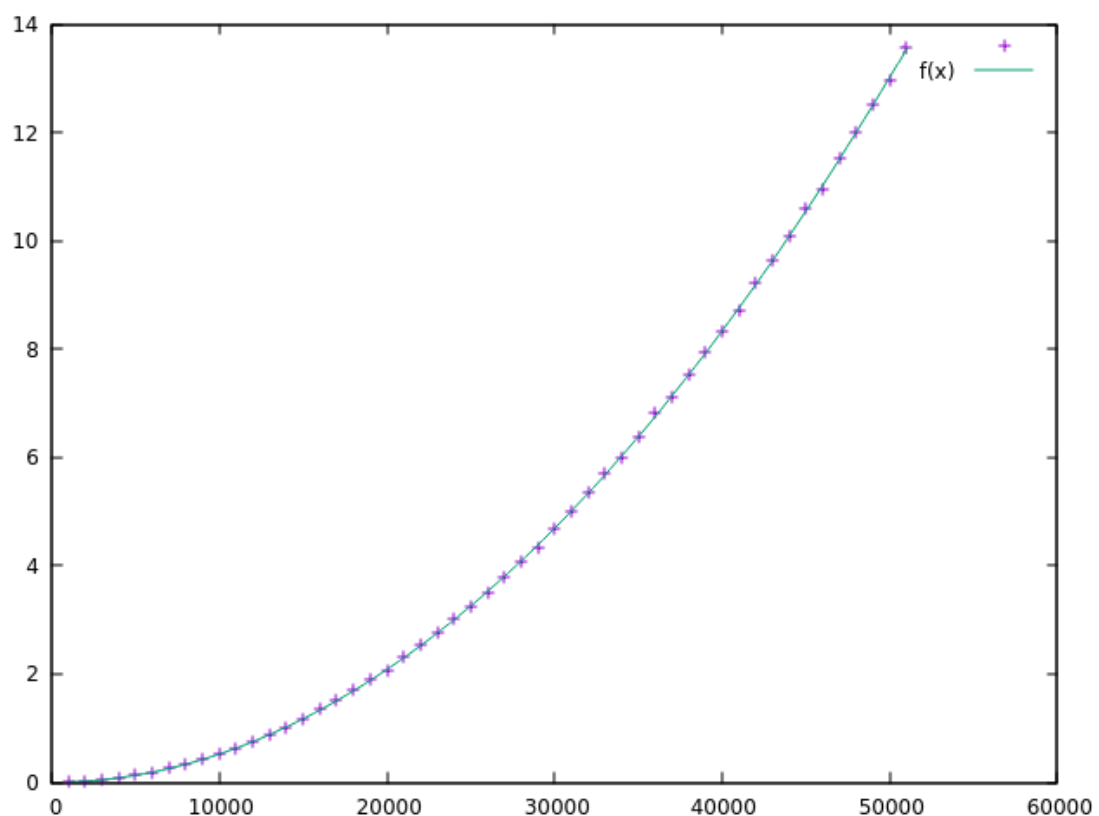
Para la medición de los tiempos, se ha usado un método que mide el tiempo antes y después de cada una de las ejecuciones con diferentes tamaños para el 'n'.

Para el cálculo de grafica han sido tomadas una serie de medidas de tiempo, concretamente unas medidas para n entre 1000 y 51000.

Para esto se ha usado una función $f(x)=a_0*(n^2)$ obteniendo un error bastante bajo de:

$$a_0 = 5.20917e-09 \quad +/- \quad 3.027e-12 \quad (0.05811\%)$$

Representación Gráfica



Como se puede observar en la gráfica, la función se adapta bastante bien a la crecida de los datos pero con algunas pequeñas variaciones, pero no tan pronunciadas como en otros algoritmos.
