Algoritmo Inserción

Algoritmo natural usado para ordenadar n elementos de numerados de forma arbitraria. Requiere $O(n^2)$ operaciones para ordenar una lista de n elementos.

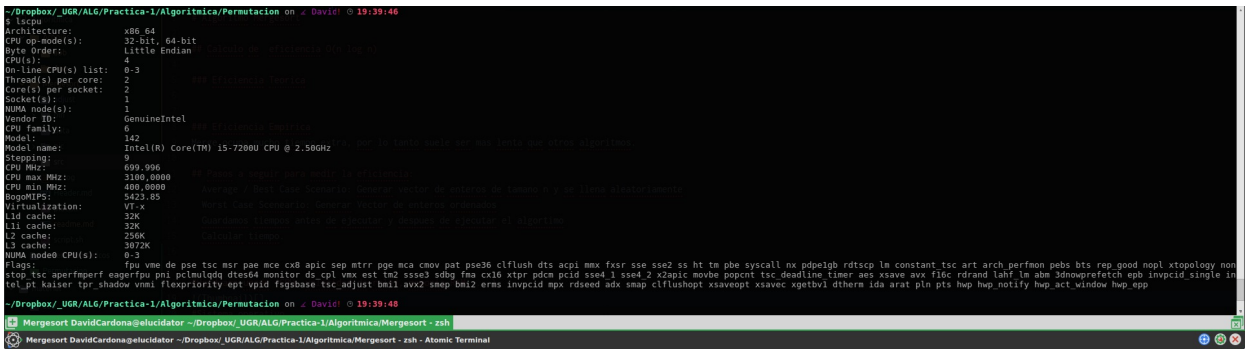
Inicialmente se tiene un solo elemento, que obviamente es un conjunto ordenado. Después, cuando hay k elementos ordenados de menor a mayor, se toma el elemento k+1 y se compara con todos los elementos ya ordenados, deteniéndose cuando se encuentra un elemento menor (todos los elementos mayores han sido desplazados una posición a la derecha) o cuando ya no se encuentran elementos (todos los elementos fueron desplazados y este es el más pequeño). En este punto se inserta el elemento k+1 debiendo desplazarse los demás elementos.

Pasos a seguir para medir la eficiencia:

1. Average: Generar vector de enteros aleatorio
2. Best Case Scenario: Generar vector de enteros ordenado
3. Worst Case Sceneario: Generar vector de enteros ordenados al reves
4. Guardamos tiempos antes de ejecutar y despues de ejecutar el algortimo
5. Calcular tiempo.

Información

Hardware usado (CPU, velocidad de reloj, memoria RAM, ...)



Compilador utilizado y opciones de compilación

```
gcc -v
Using built-in specs.
```

```
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/5/lto-wrapper
Target: x86_64-linux-gnu
gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.9)
```

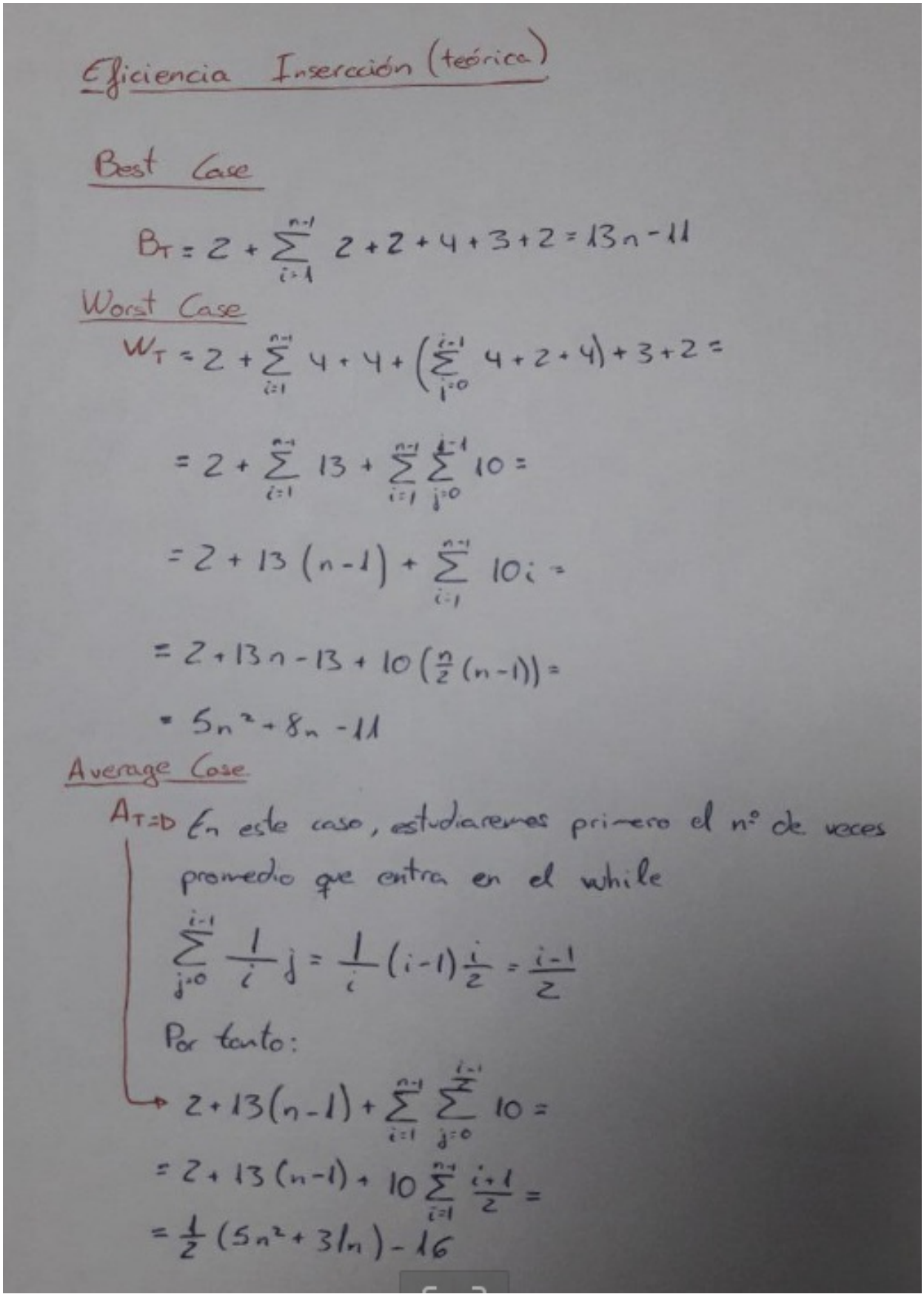
Compilación

```
g++ -std=c++11 ./src/$PROGRAMA.cpp -o ./bin/$PROGRAMA

Usamos el siguiente script:
# Variables:
PROGRAMA=$1
SALIDA=./data/tiempo_best_case_$1.dat
MENSAJE_INICIO="Se inicia la ejecución del algoritmo $1:"
MENSAJE_FINAL="Fin de la ejecución. Se ha creado un fichero con los resultados."

# Se genera el ejecutable con el algoritmo de ordenación:
g++ -std=c++11 ./src/$PROGRAMA.cpp -o ./bin/$PROGRAMA
echo "$MENSAJE_INICIO"

# Variables:
INICIO=1000
FIN=30000
INCREMENTO=1000

i=$INICIO
echo > $SALIDA
while [ $i -le $FIN ]
do
    echo Vector size = $i
    echo "`./bin/$PROGRAMA $i 10000`" >> $SALIDA
    i=$((i+$INCREMENTO))
done

rm -fr ./bin/$PROGRAMA

echo "$MENSAJE_FINAL"
```

Sistema operativo

Desarrollo completo del cálculo de la eficiencia teórica y gráfica.

Calculo de eficiencia O(n^2)



Eficiencia Inserción (teórica)

Best Case

$$B_T = 2 + \sum_{i=1}^{n-1} 2+2+4+3+2 = 13n - 11$$

Worst Case

$$W_T = 2 + \sum_{i=1}^{n-1} 4+4+\left(\sum_{j=0}^{i-1} 4+2+4\right)+3+2 =$$

$$= 2 + \sum_{i=1}^{n-1} 13 + \sum_{i=1}^{n-1}\sum_{j=0}^{i-1} 10 =$$

$$= 2 + 13(n-1) + \sum_{i=1}^{n-1} 10i =$$

$$= 2 + 13n - 13 + 10\left(\frac{n}{2}(n-1)\right) =$$

$$= 5n^2 + 8n - 11$$

Average Case

A_T :D En este caso, estudiaremos primero el nº de veces promedio que entra en el while

$$\sum_{j=0}^{i-1} \frac{1}{i} j = \frac{1}{i}(i-1)\frac{i}{2} = \frac{i-1}{2}$$

Por tanto:

$$2 + 13(n-1) + \sum_{i=1}^{n-1}\sum_{j=0}^{\frac{i-1}{2}} 10 =$$

$$= 2 + 13(n-1) + 10\sum_{i=1}^{n-1} \frac{i+1}{2} =$$

$$= \frac{1}{2}(5n^2 + 31n) - 16$$

Eficiencia Empirica

Hemos tomado multiples medidas y sobre esto hemos realizado los ajustes y graficas.

Ejemplo de medidas Worst Case

El algoritmo de inserción es cuadratico junto con los otros dos, burbuja y selección pero aun asi hay diferencias en este caso el numero de comparaciones e intercambios que se hacen.

N Elementos	Tiempo
1000	0.00487127
2000	0.0175072
3000	0.028667
4000	0.063623
5000	0.0773883
6000	0.111883
7000	0.142794
8000	0.187937
9000	0.247467
10000	0.299123
11000	0.373189
12000	0.418797
13000	0.50118
14000	0.583927
15000	0.662768
16000	0.75421
17000	0.852025

18000	0.948132
19000	1.07889
20000	1.18116
21000	1.31908
22000	1.44115
23000	1.58941
24000	1.70263
25000	1.90796
26000	2.0077
27000	2.17487
28000	2.34309
29000	2.47959
30000	2.71084

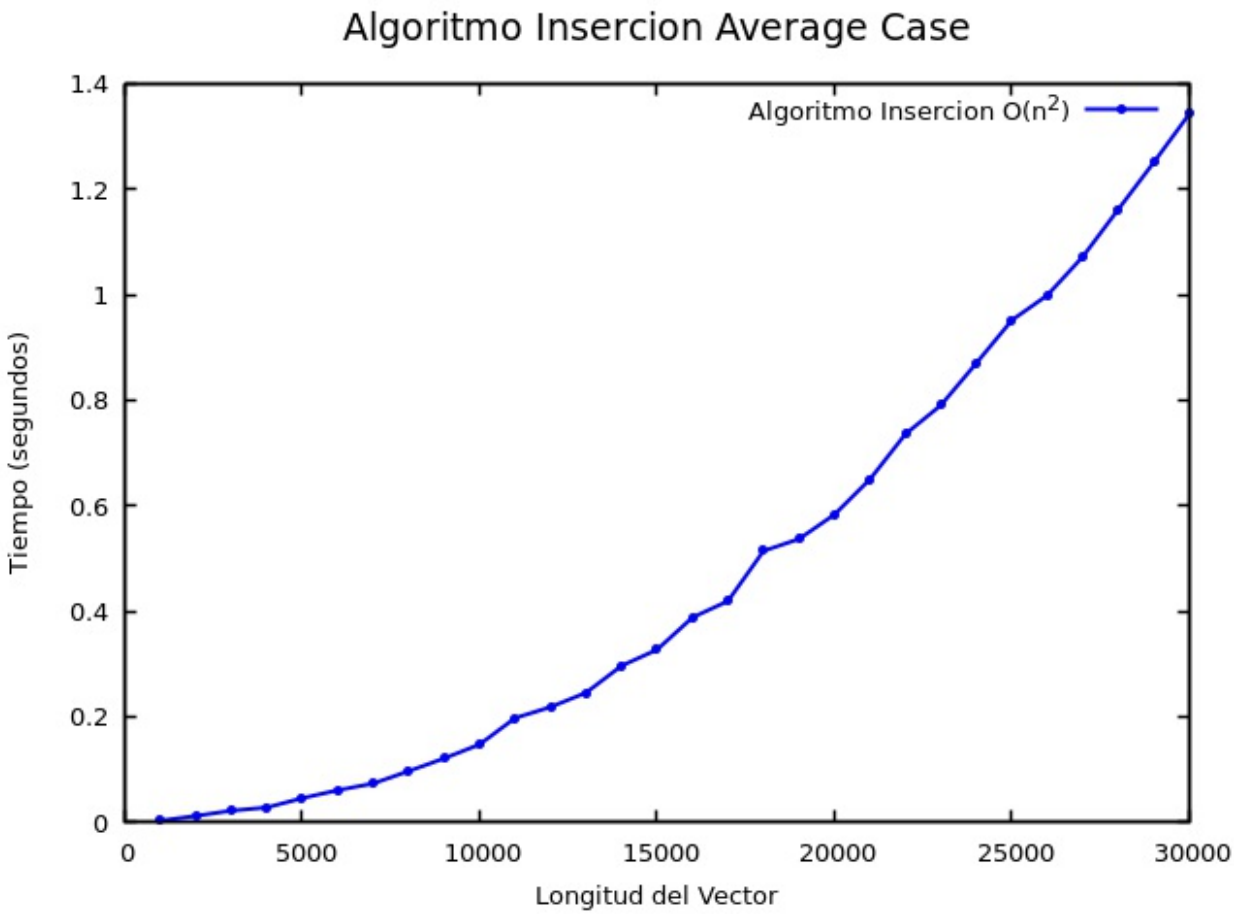
Parámetros usados para el cálculo de la eficiencia empírica y gráfica.

Para el calculo de las gráficas hemos usado el script:

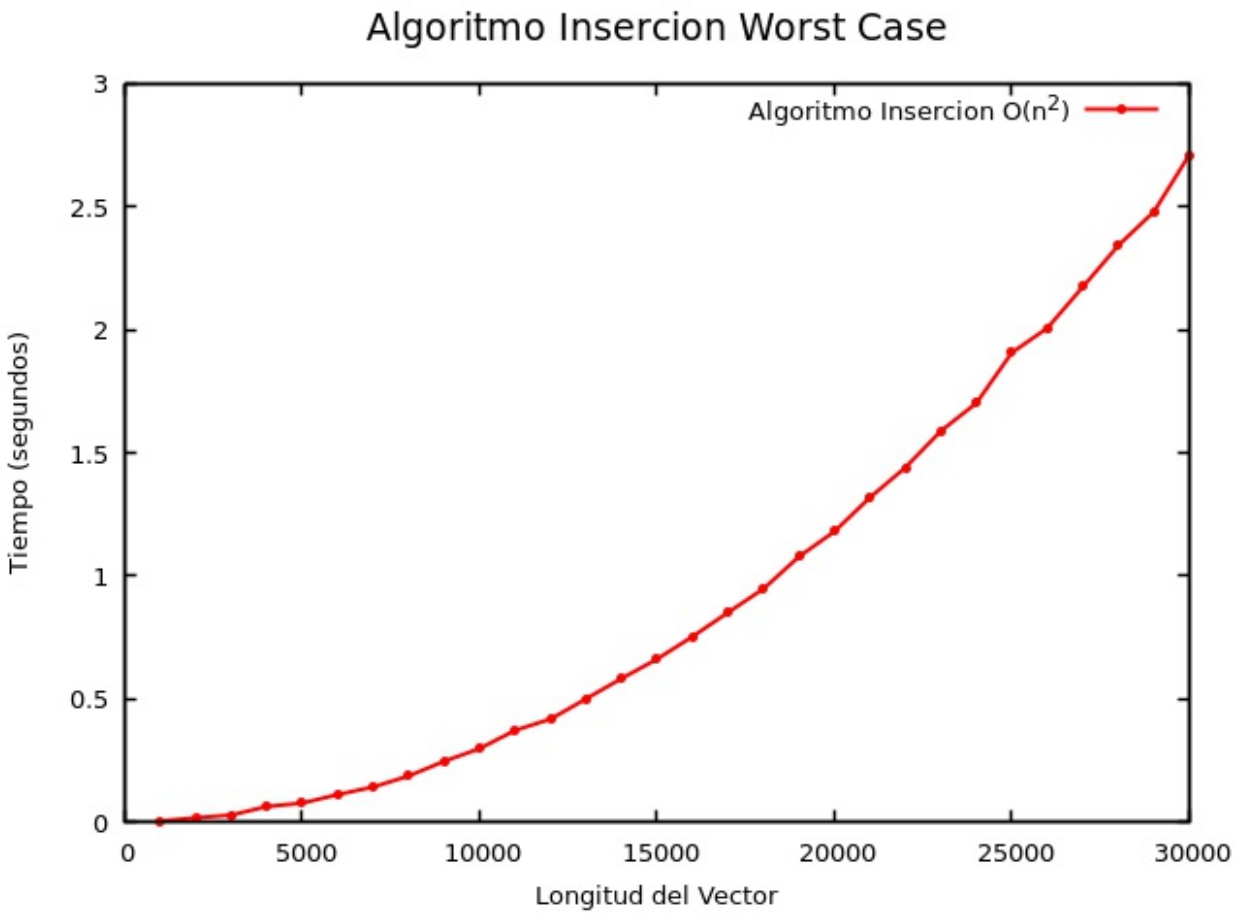
```
#!/bin/bash

#Variables:
OUTPUT=./data/grafica_tiempo_average_case_insercion.png
TITULO="Algoritmo Insercionn Average Case"
XLABEL="Longitud del Vector"
YLABEL="Tiempo (segundos)"
LEYENDA="Algoritmo Insercionn O(n^2)"
FICHERO_DATOS="./data/tiempo_average_case_insercion.dat"
COLOR=blue

gnuplot<<FIN
# Terminal para png:
set terminal pngcairo enhanced font 'Verdana,10'
set border linewidth 1.5
```

```
# Estilo de línea y color:
set style line 1 lc rgb '$COLOR' lt 1 lw 2 pt 7 pi 0 ps 0.5
set pointintervalbox 0

# Nombre de la imagen resultante:
set output '$OUTPUT'

# Título y ejes:
set title "$TITULO" enhanced font 'Verdana,14'
set xlabel "$XLABEL"
set ylabel "$YLABEL"

set autoscale

plot "$FICHERO_DATOS" title '$LEYENDA' with linespoints ls 1
FIN
```

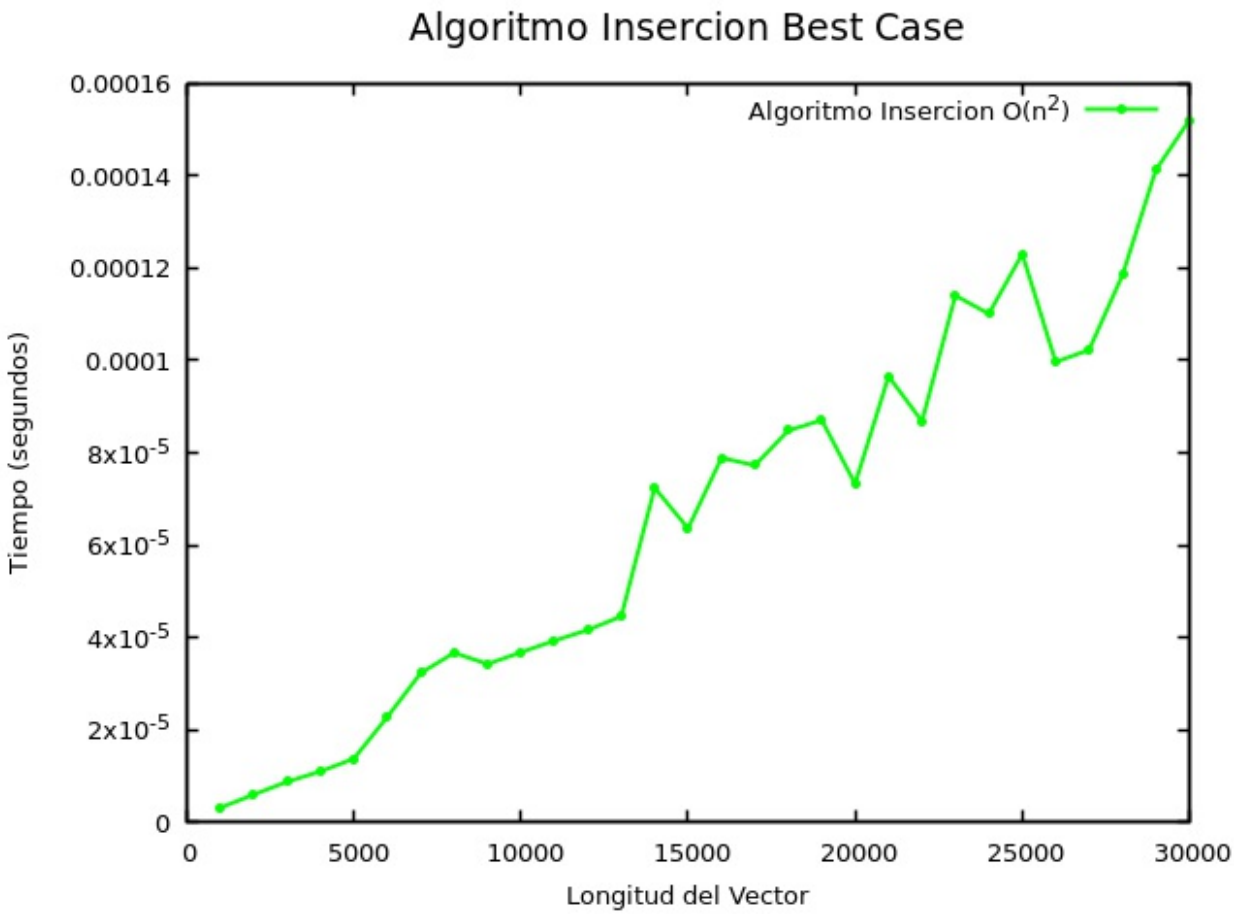
Average


Algoritmo Insercion Average Case
Algoritmo Insercion O(n^2)
Tiempo (segundos)
Longitud del Vector
0
0.2
0.4
0.6
0.8
1
1.2
1.4
0
5000
10000
15000
20000
25000
30000

Worst Case

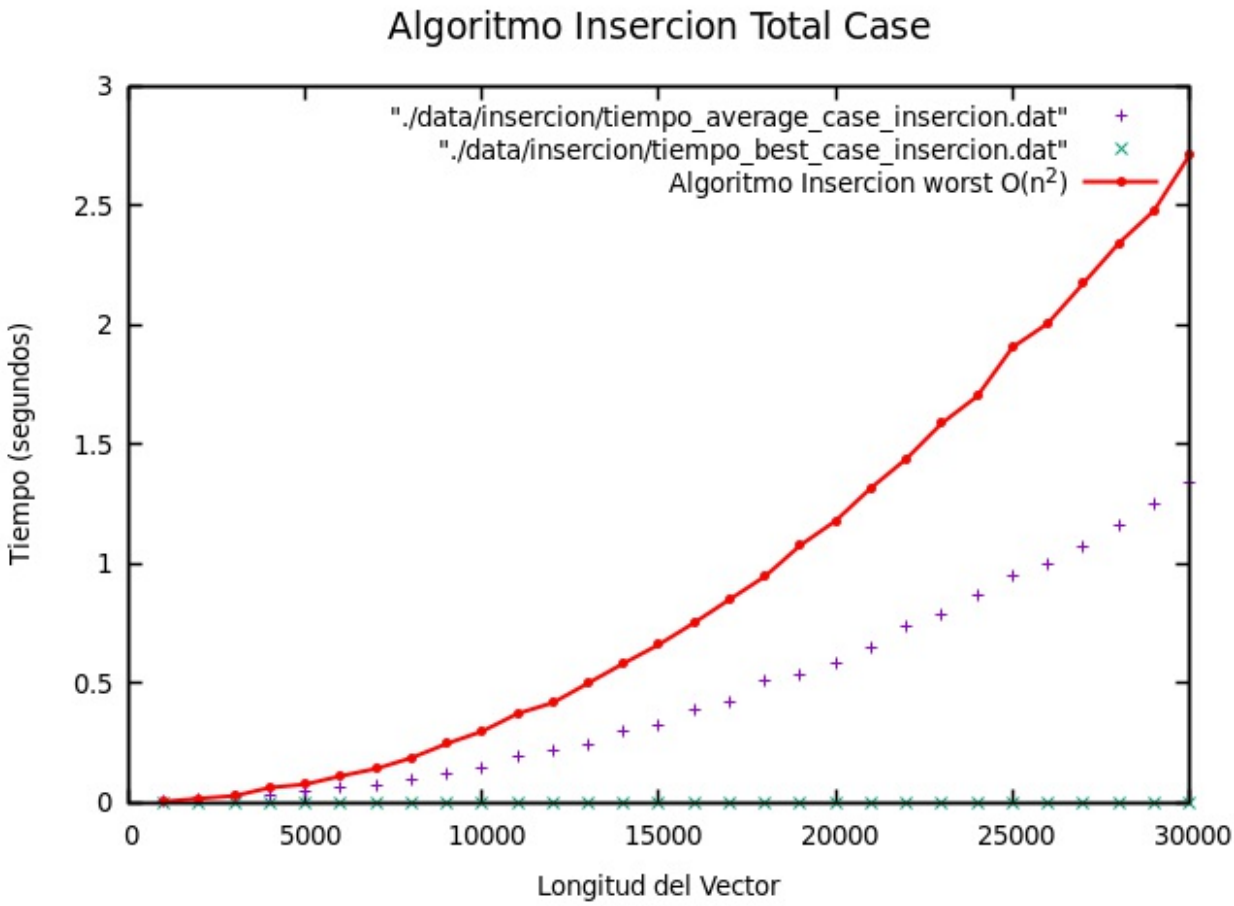

Algoritmo Insercion Worst Case
Algoritmo Insercion O(n^2)
Tiempo (segundos)
Longitud del Vector
0
0.5
1
1.5
2
2.5
3
0
5000
10000
15000
20000
25000
30000

Best Case

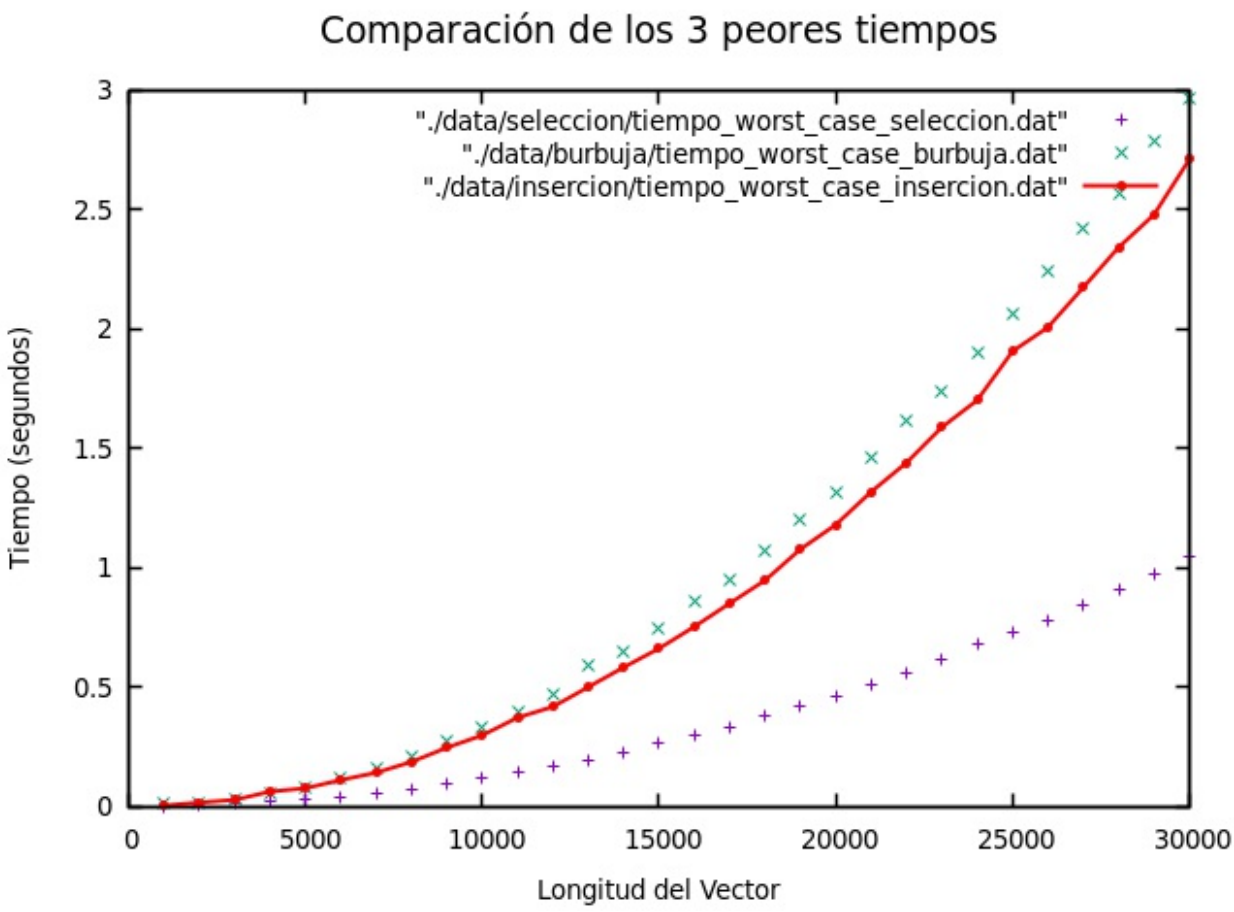

Algoritmo Insercion Best Case
Algoritmo Insercion O(n^2)
Tiempo (segundos)
Longitud del Vector
0.00016
0.00014
0.00012
0.0001
8x10^-5
6x10^-5
4x10^-5
2x10^-5
0
0
5000
10000
15000
20000
25000
30000

Ajuste de la curva teórica a la empírica: mostrar resultados del ajuste y gráfica.

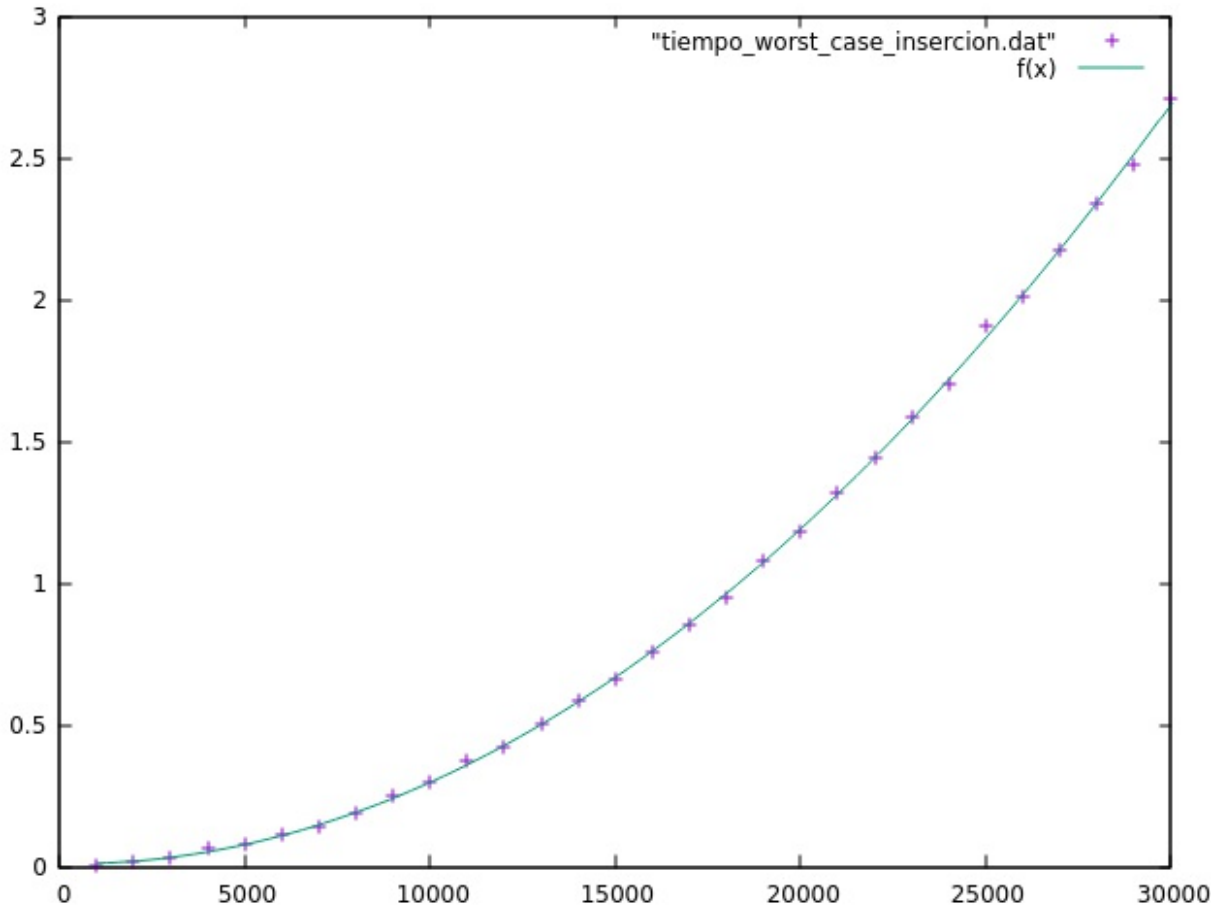
Grafica de comparacion con todos los casos



Comparacion con los otros dos algoritmos



Ajuste



Ajuste

```
*******************************************************************************
Mon Mar  5 19:10:36 2018


FIT:    data read from "data/insercion/tiempo_worst_case_insercion.dat"
        format = z
        #datapoints = 30
        residuals are weighted equally (unit weight)

function used for fitting: f(x)
  f(x)=a*x**2 + b*x + c
fitted parameters initialized with current variable values

iter      chisq       delta/lim  lambda   a             b             c
   0 1.7515977662e+01   0.00e+00  2.80e-01    1.158524e-09   3.538991e-08   9.31117
   6 4.9433033151e-03  -2.06e-02  2.80e-07    3.034647e-09  -1.675779e-06   9.95043

After 6 iterations the fit converged.
```

```
final sum of squares of residuals : 0.0049433
rel. change during last iteration : -2.06321e-07

degrees of freedom    (FIT_NDF)                        : 27
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 0.0135309
variance of residuals (reduced chisquare) = WSSR/ndf   : 0.000183085

Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a               = 3.03465e-09      +/- 3.693e-11    (1.217%)
b               = -1.67578e-06     +/- 1.18e-06     (70.41%)
c               = 0.00995044       +/- 0.007934     (79.74%)

correlation matrix of the fit parameters:
                a      b      c
a               1.000
b              -0.970  1.000
c               0.770 -0.882  1.000
```