Algoritmo Selección

El algoritmo de selección utilizado para encontrar el k-ésimo menor número de un vector.

El caso más simple de un algoritmo de selección es contrar el mínimo o máximo de entre los elementos de un vector. Para este y otros ejemplos, estudiamos 3 posibilidades:

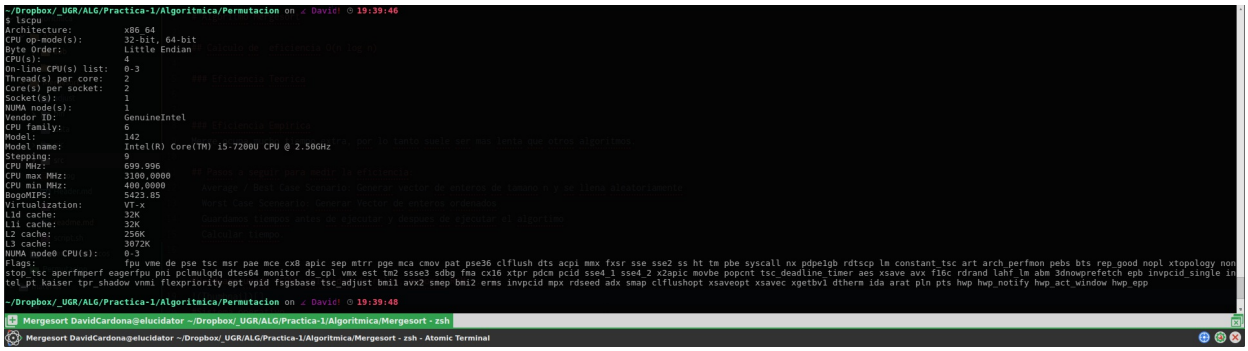
1.En el mejor caso en el que el vector ya este ordenado(para el caso de ordenación de menor a mayor, de 1 a 10, por ejemplo). 2.En el caso promedio simplemente encontramos el vector lleno de elementos random(para el caso anterior sería una descripción aceptable también). 3.En el peor caso el vecto se encontraría ordenado completamente de forma opuesta a lo que buscamos(Para nuestro ejemplo de antes, sería de 10-1).

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)
```

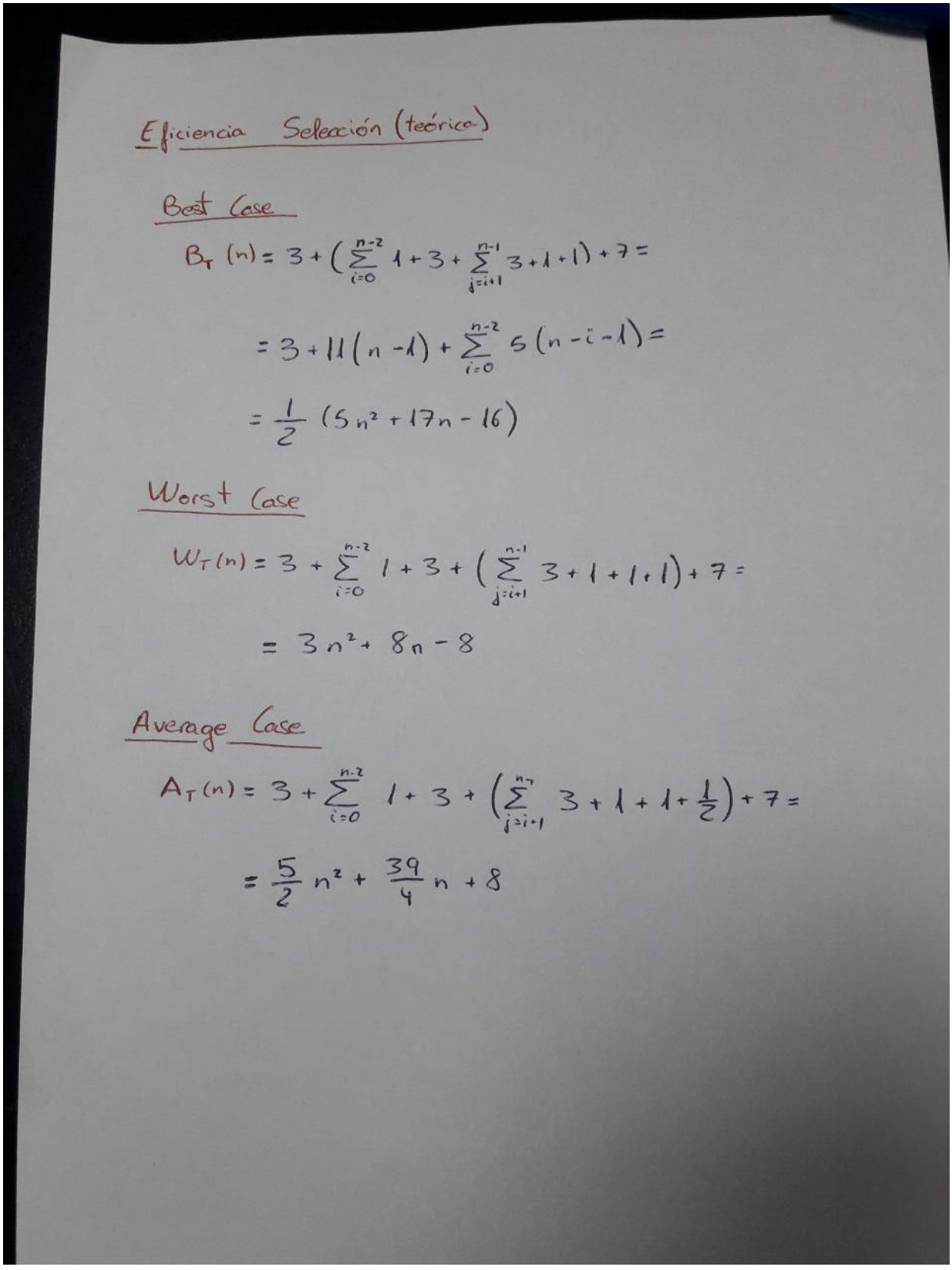
Compilación

```
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 Selección (teórica)

Best Case

$$B_T(n) = 3 + \left(\sum_{i=0}^{n-2} 1+3+\sum_{j=i+1}^{n-1} 3+1+1\right)+7 =$$

$$= 3 + 11(n-1) + \sum_{i=0}^{n-2} 5(n-i-1) =$$

$$= \frac{1}{2}(5n^2+17n-16)$$

Worst Case

$$W_T(n) = 3 + \sum_{i=0}^{n-2} 1+3+\left(\sum_{j=i+1}^{n-1} 3+1+1+1\right)+7 =$$

$$= 3n^2+8n-8$$

Average Case

$$A_T(n) = 3 + \sum_{i=0}^{n-2} 1+3+\left(\sum_{j=i+1}^{n-1} 3+1+1+\frac{1}{2}\right)+7 =$$

$$= \frac{5}{2}n^2+\frac{39}{4}n+8$$

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	0.001194
2000	0.00542
3000	0.011826
4000	0.020661
5000	0.031669
6000	0.042889
7000	0.057995
8000	0.075118
9000	0.094499
10000	0.117745
11000	0.140843
12000	0.167326
13000	0.196081
14000	0.227455
15000	0.263979
16000	0.297902

17000	0.334467
18000	0.376796
19000	0.420365
20000	0.463759
21000	0.512066
22000	0.561706
23000	0.612677
24000	0.678095
25000	0.73032
26000	0.782029
27000	0.845572
28000	0.908734
29000	0.972301
30000	1.04675

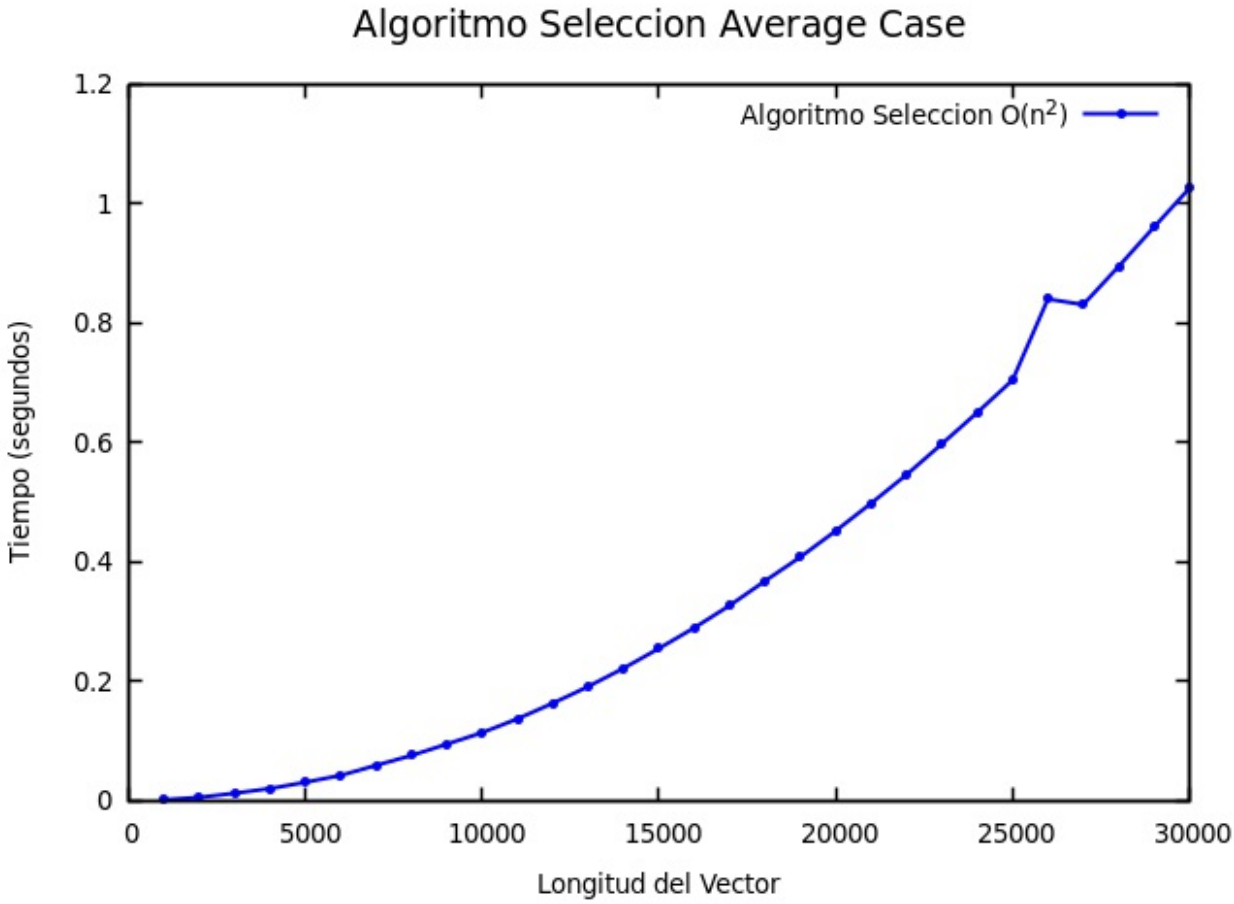
Parámetros usados para el cálculo de la eficiencia empírica y gráfica.

Para el calculo de las gráficas hemos usado el script:

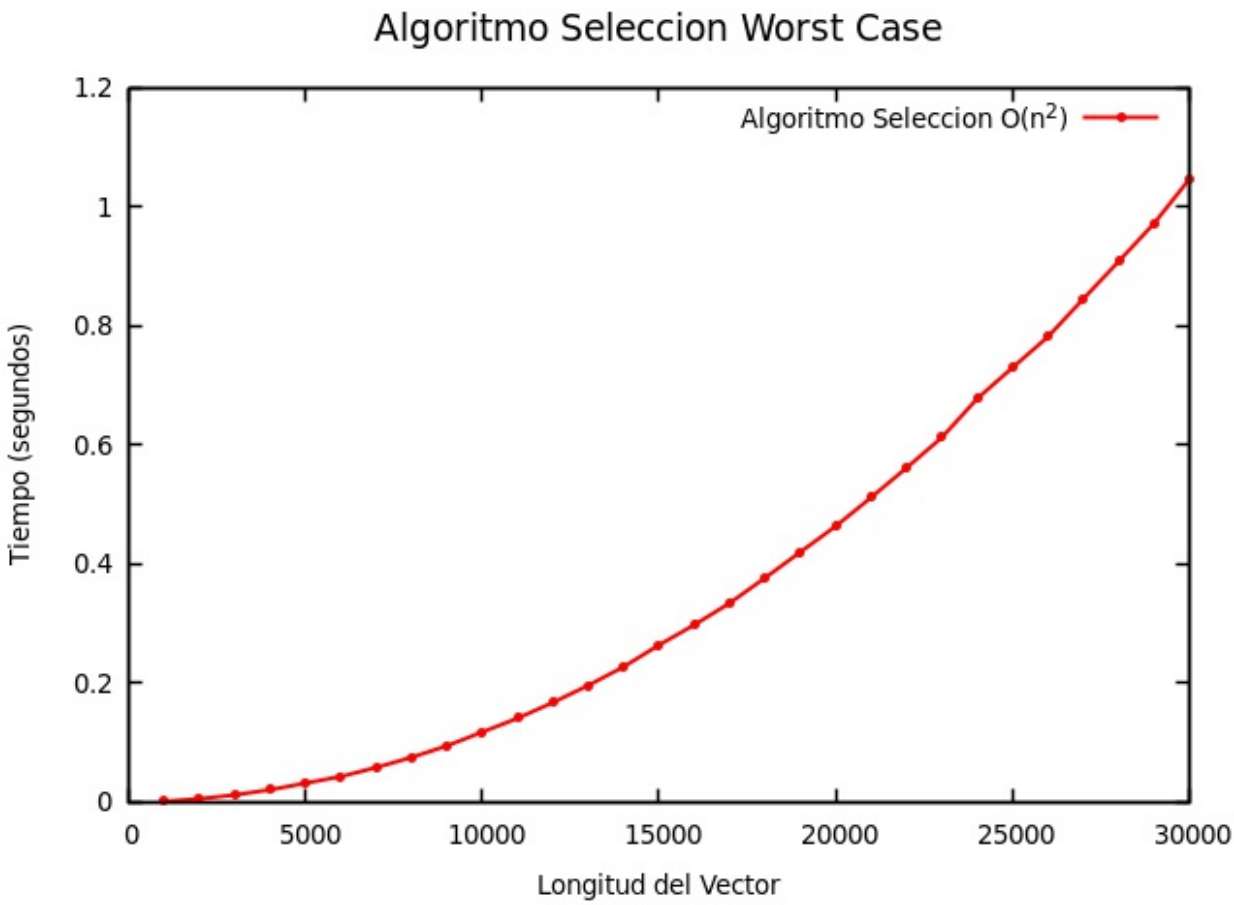
```
#!/bin/bash

#Variables:
OUTPUT=./data/grafica_tiempo_average_case_seleccion.png
TITULO="Algoritmo Seleccion Average Case"
XLABEL="Longitud del Vector"
YLABEL="Tiempo (segundos)"
LEYENDA="Algoritmo Seleccion O(n^2)"
FICHERO_DATOS="./data/tiempo_average_case_seleccion.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

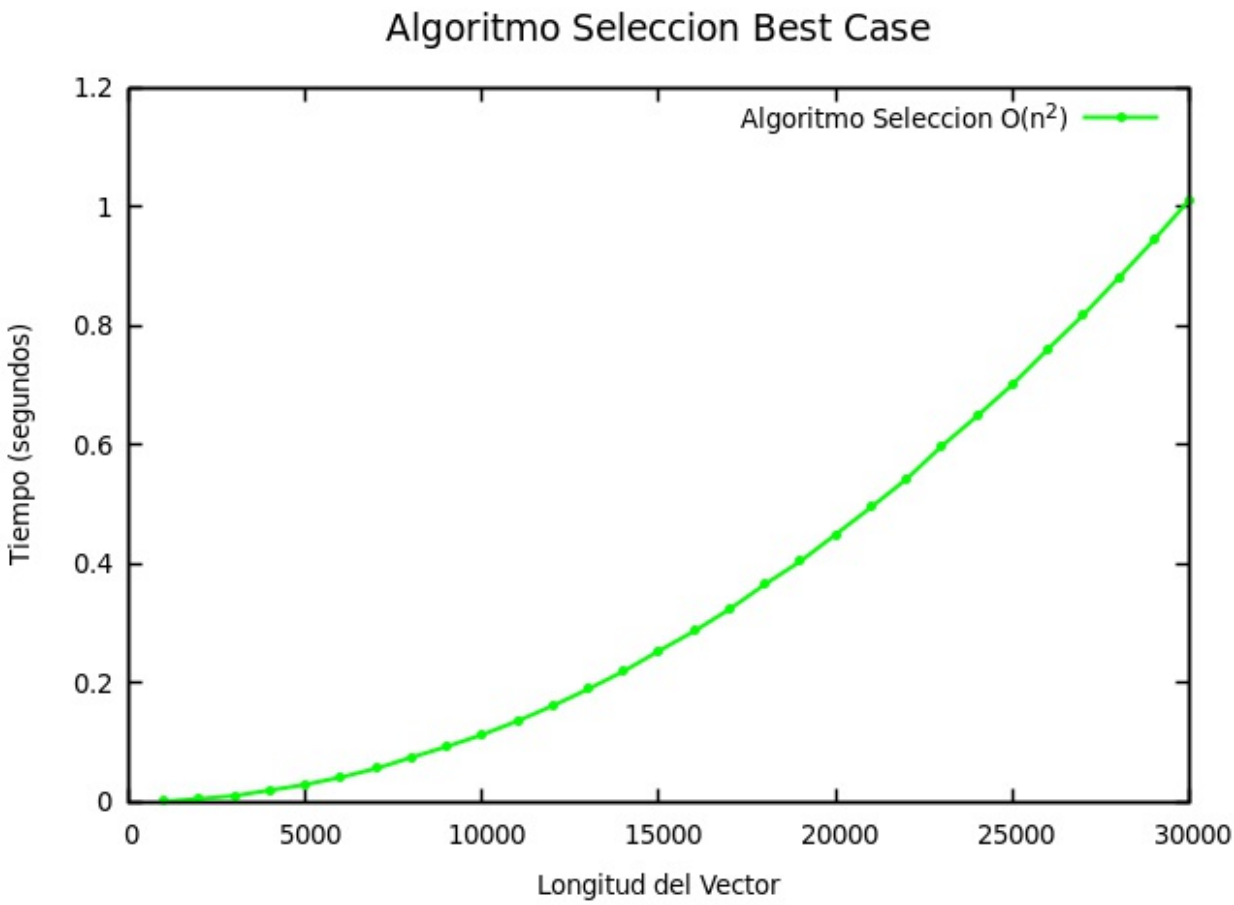

Algoritmo Seleccion Average Case
Algoritmo Seleccion O(n^2)
Tiempo (segundos)
Longitud del Vector
0
0.2
0.4
0.6
0.8
1
1.2
0
5000
10000
15000
20000
25000
30000

Worst Case


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

)

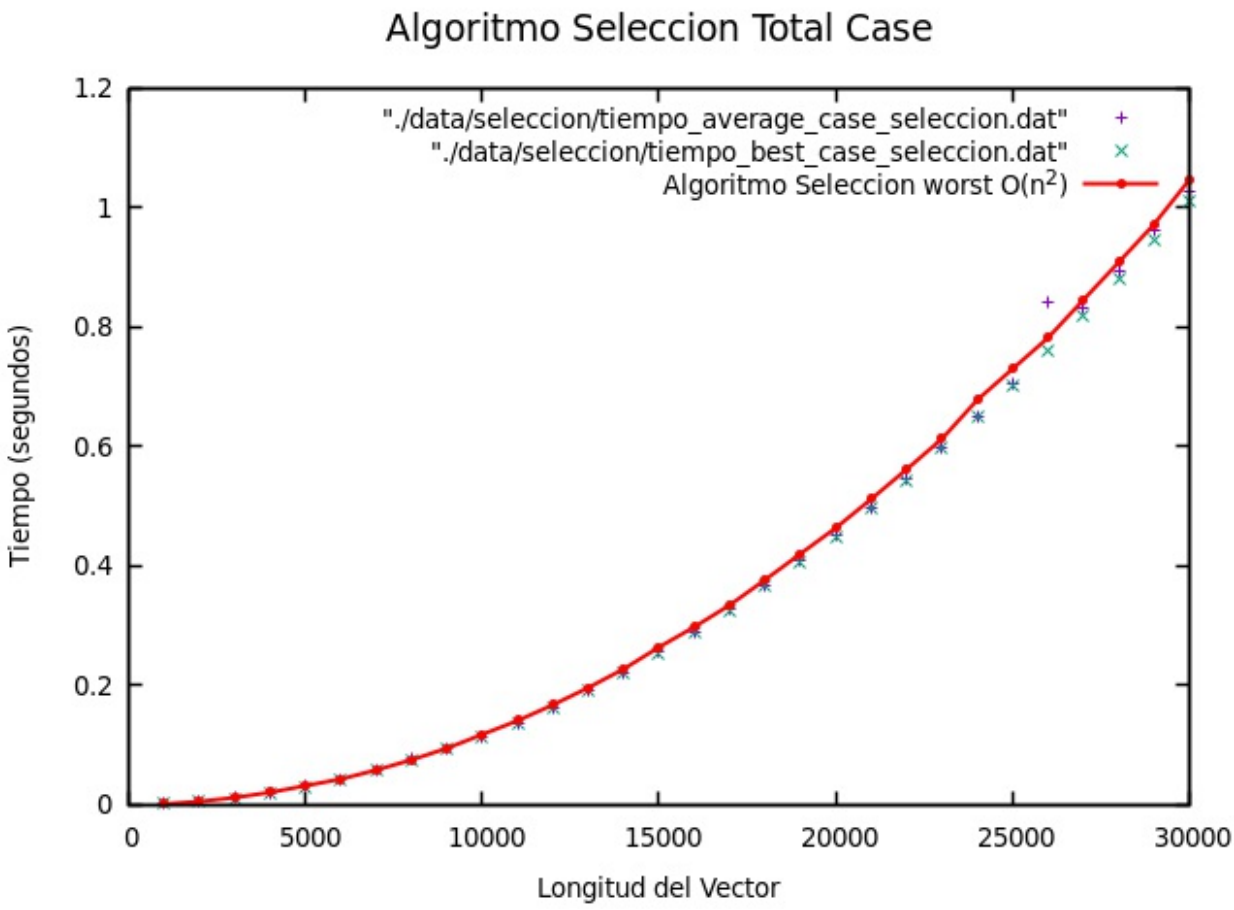
Best Case


Algoritmo Seleccion Best Case
Algoritmo Seleccion O(n2)
Tiempo (segundos)
Longitud del Vector
0
0.2
0.4
0.6
0.8
1
1.2
0
5000
10000
15000
20000
25000
30000

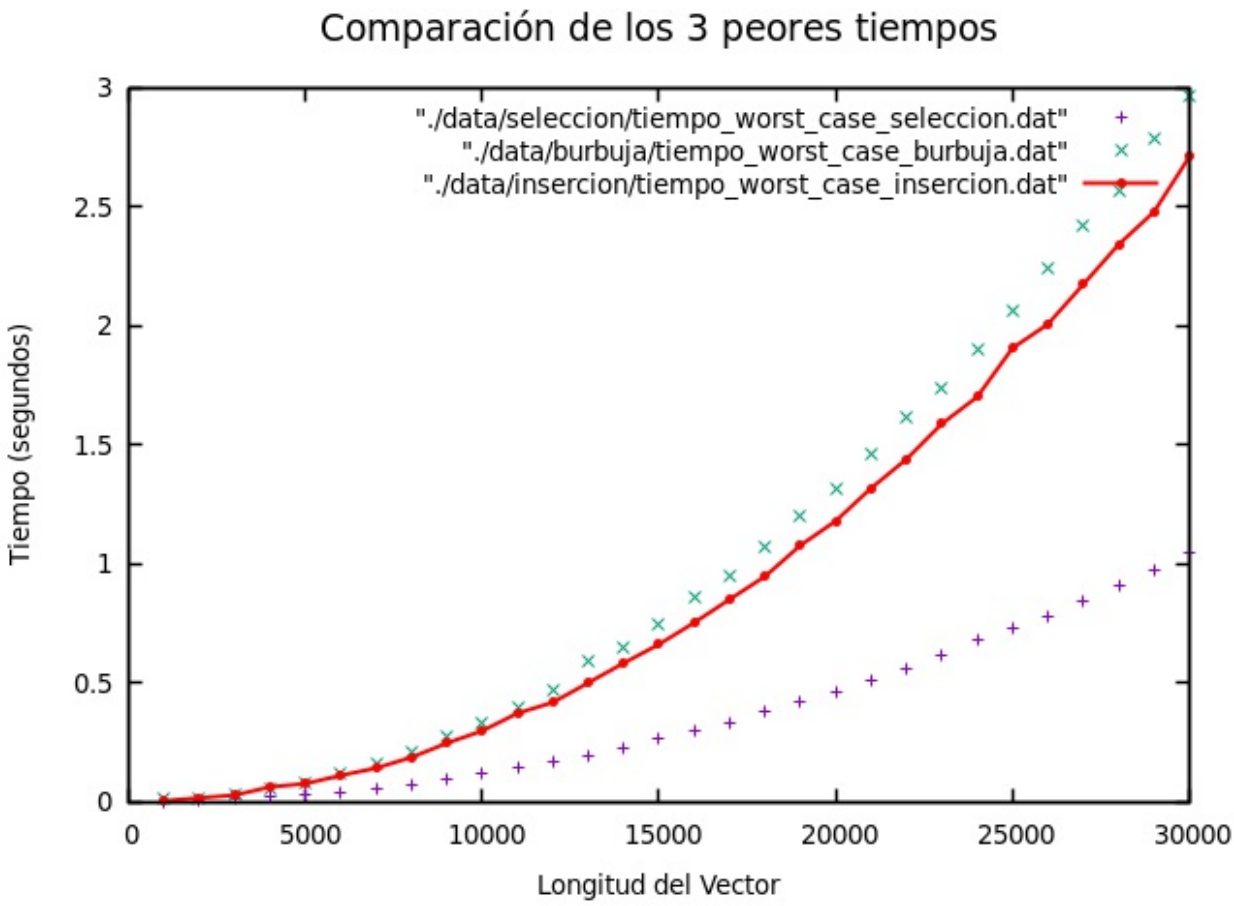
)

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

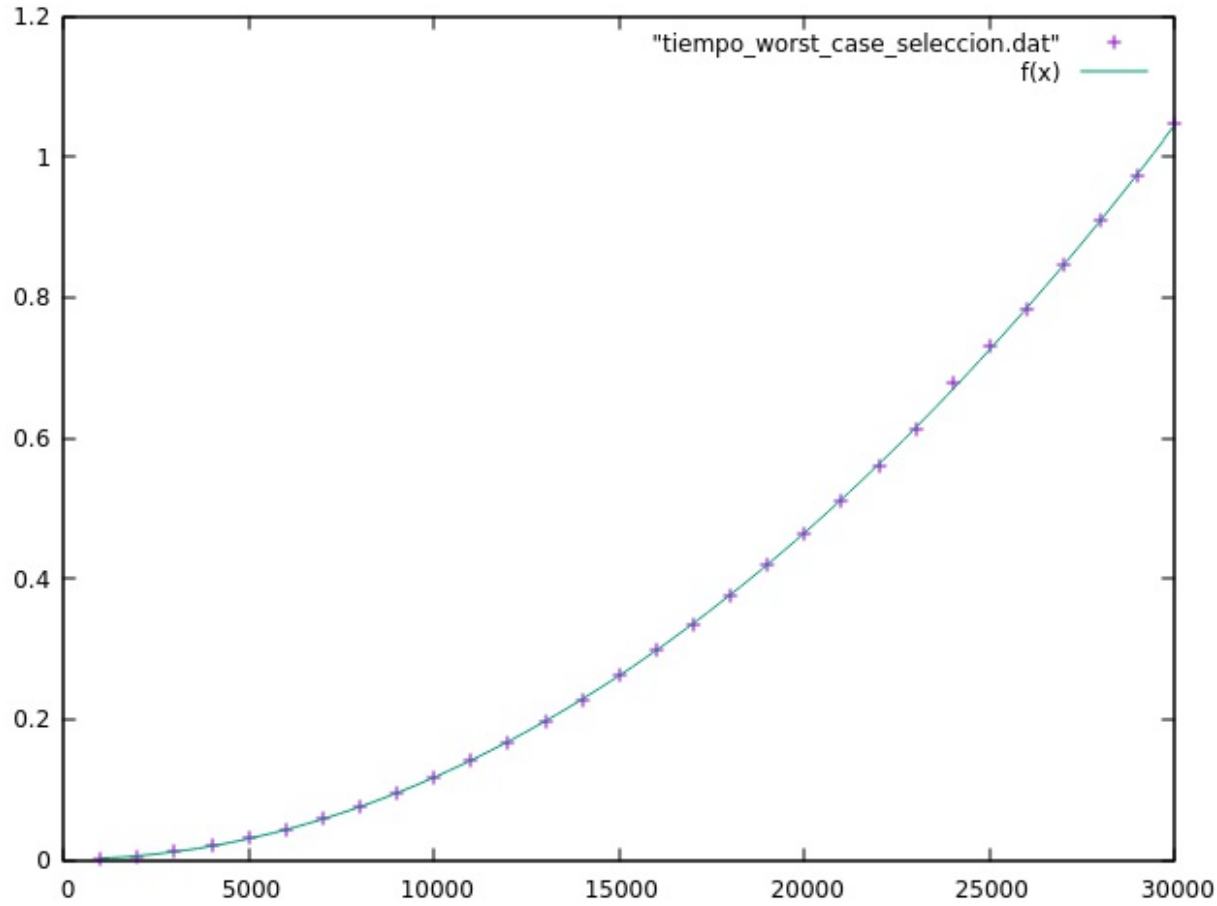
Grafica de comparacion con todos los casos



Comparacion con los otros dos algoritmos



Ajuste



Ajuste

```
*******************************************************************************
Sun Mar 18 18:18:27 2018


FIT:    data read from "tiempo_worst_case_seleccion.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



*******************************************************************************
Sun Mar 18 18:18:51 2018
```

```
FIT:    data read from "tiempo_worst_case_seleccion.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 5.2744314661e+18   0.00e+00  2.42e+08    1.000000e+00   1.000000e+00   1.00000
  12 1.5508768835e-04  -1.31e-02  2.42e-04    1.158525e-09   3.538594e-08   9.31146

After 12 iterations the fit converged.
final sum of squares of residuals : 0.000155088
rel. change during last iteration : -1.31004e-07

degrees of freedom    (FIT_NDF)                       : 27
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 0.00239666
variance of residuals (reduced chisquare) = WSSR/ndf   : 5.74399e-06

Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a               = 1.15852e-09      +/- 6.541e-12    (0.5646%)
b               = 3.53859e-08      +/- 2.09e-07     (590.6%)
c               = 0.000931146      +/- 0.001405     (150.9%)

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