PRÁCTICA 1:

EFICIENCIA

Laura Tirado López
María del Mar García Cabello

Práctica 1: Eficiencia

EJERCICIO 1

El algoritmo de ocurrencias es de orden O(n).

```
////////////////////////////////////////////////////
// CONTAR OCURRENCIAS PALABRA
////////////////////////////////////////////////////
cout << "Contar ocurrencias (con toma de tiempos) " << endl;


for (int fin = 10; fin < 100000 ; fin+= 1000){    // O(n)
string b="hidalgo";  // O(1)

start = clock();  // O(1)
for (int iteraciones = 0; iteraciones < 1000; iteraciones++)  // O(n)
     pos = contar_hasta(Q, 0,fin, b); // O(n)
end= clock(); // O(1)
double dif = end-start;  // O(1)
cout << fin << "     " << dif/(double) (CLOCKS_PER_SEC * 1000.0) << endl;  // O(1)
}
```

TERMINAL:

$g++ ocurrencias.cpp -o ocurrencias //Compilamos

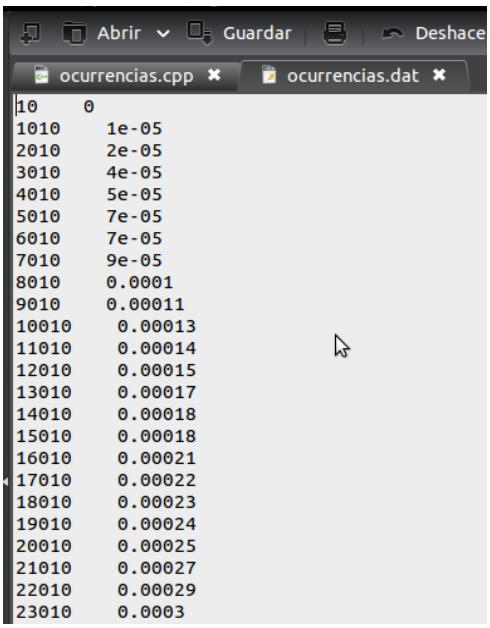
```
muscraziest@muscraziest-SVE1511C5E:~/Escritorio$ g++ ocurrencias.cpp -o ocurrencias
muscraziest@muscraziest-SVE1511C5E:~/Escritorio$ ./ocurrencias
Tamanio dic: 86162 Capacidad dic: 131072
Tamanio Quijote: 389838 Capacidad Quijote: 524288
Contar ocurrencias (con toma de tiempos)
10    0
1010    2e-05
2010    2e-05
3010    4e-05
4010    5e-05
5010    6e-05
6010    8e-05
7010    9e-05
8010    0.0001
9010    0.00012
10010    0.00012
11010    0.00014
12010    0.00016
13010    0.00016
14010    0.00018
15010    0.00019
16010    0.0002
17010    0.00022
```

$./ocurrencias //Lo ejecutamos

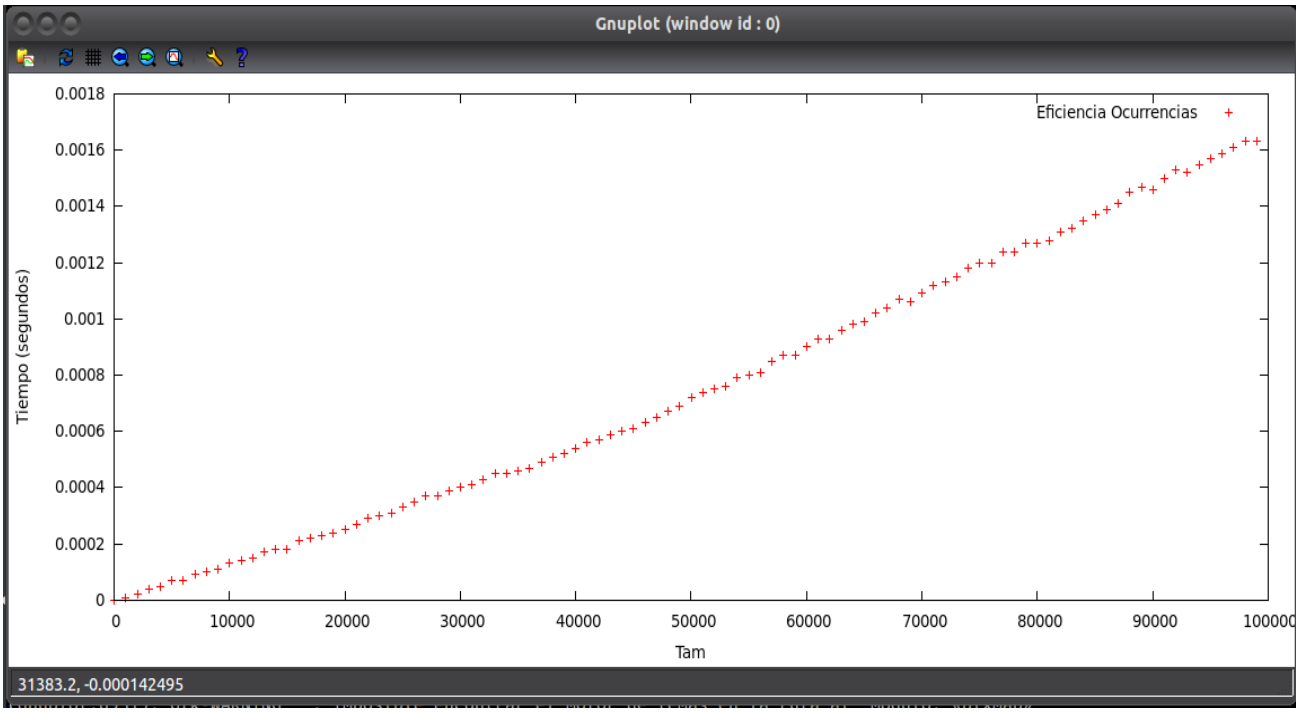
Laura Tirado López
María del Mar García Cabello

$./ocurrencias > ocurrencias.dat //Guardamos los datos

En la siguiente imagen vemos como tenemos los datos en el archivo ocurrencias.dat.



$gnuplot //Abrimos gnuplot
$gnuplot > plot 'ocurrencias.dat ' title 'Eficiencia Ocurrencias'

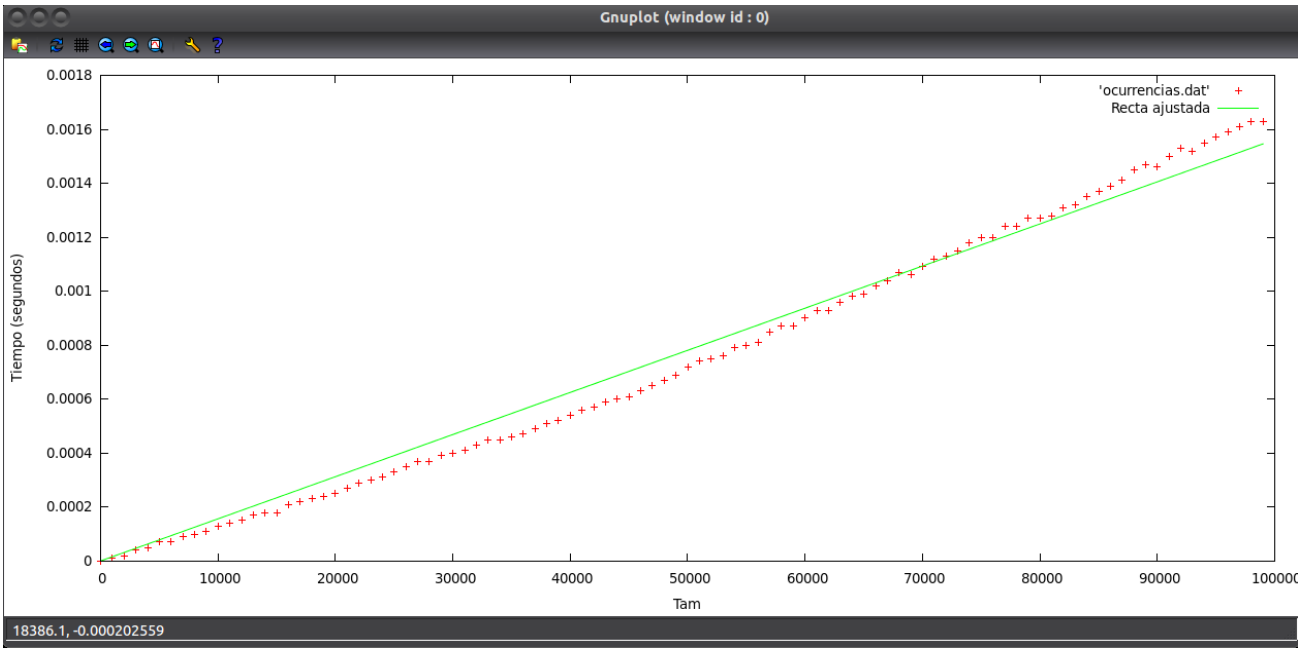


Laura Tirado López
María del Mar García Cabello

$gnuplot> f(x)=a*x //Definimos la función
$gnuplot>fit f(x) 'ocurrencias.dat' via a

```
/
 Iteration 2
 WSSR        : 3.27717e-07       delta(WSSR)/WSSR   : -0.0239583
 delta(WSSR) : -7.85156e-09      limit for stopping : 1e-05
 lambda      : 573.105

resultant parameter values

a               = 1.5616e-08
/
 Iteration 3
 WSSR        : 3.27717e-07       delta(WSSR)/WSSR   : -2.39534e-10
 delta(WSSR) : -7.84995e-17      limit for stopping : 1e-05
 lambda      : 57.3105

resultant parameter values

a               = 1.5616e-08

After 3 iterations the fit converged.
final sum of squares of residuals : 3.27717e-07
rel. change during last iteration : -2.39534e-10

degrees of freedom    (FIT_NDF)                        : 99
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 5.7535e-05
variance of residuals (reduced chisquare) = WSSR/ndf   : 3.31027e-09

Final set of parameters            Asymptotic Standard Error
=======================            ==========================

a               = 1.5616e-08       +/- 1.004e-10    (0.6429%)


correlation matrix of the fit parameters:

                a
a                1.000
```

$gnuplot> plot 'ocurrencias.dat', f(x) title 'Recta ajustada'



Laura Tirado López
María del Mar García Cabello

En esta ultima imagen tendríamos la correlación entre los datos de ocurrencias.dat y la recta f(x)=1,5616e-08 x.

Laura Tirado López
María del Mar García Cabello

EJERCICIO 2

a) Burbuja

Algoritmo de ordenación burbuja:

```
/* algoritmo de ordenacion por burbuja
 T: vector sobre el que se lee el fichero
 inicial: primera posicion desde la que buscar
 final : posicion siguiente a la ultima para buscar (desde V[0] hasta V[fin-1])
 */
void burbuja(vector<string> & T, int inicial, int final) {
int i, j;
string aux;
for (i = inicial; i < final - 1; i++)
        for (j = final - 1; j > i; j--)
                if (T[j] < T[j-1])
                {
                        aux = T[j];
                        T[j]= T[j-1];
                        T[j-1] = aux;
                }

}
```

a.1) Análisis de eficiencia teórico

$$\sum_{i=inicio}^{final-2}\sum_{j=i+1}^{final-1} a = \sum_{i=1}^{n-2}\sum_{j=i+1}^{n-1} a = \sum_{i=1}^{n-2} a(n-i-1) = a\sum_{i=1}^{n-2} n - \sum_{i=1}^{n-2} i - \sum_{i=1}^{n-2} 1 = \frac{an^2-3an+2a}{2} = \frac{a}{2}n^2 - \frac{3a}{2}n + a$$

El algoritmo es de orden 0(n²).

a.2) Análisis de eficiencia práctico

-Entrada: lema.txt

TERMINAL:

$g++ ordenacion.cpp -o burbuja
$./burbuja

Laura Tirado López
María del Mar García Cabello

```
100 0
5100 0.41
10100 1.71
15100 3.9
20100 6.93
25100 10.69
30100 15.24
35100 22.85
40100 33.18
45100 42.59
50100 54.02
55100 67.71
60100 86.41
65100 105.6
70100 132.56
75100 135.94
80100 151.93
85100 172.92
```

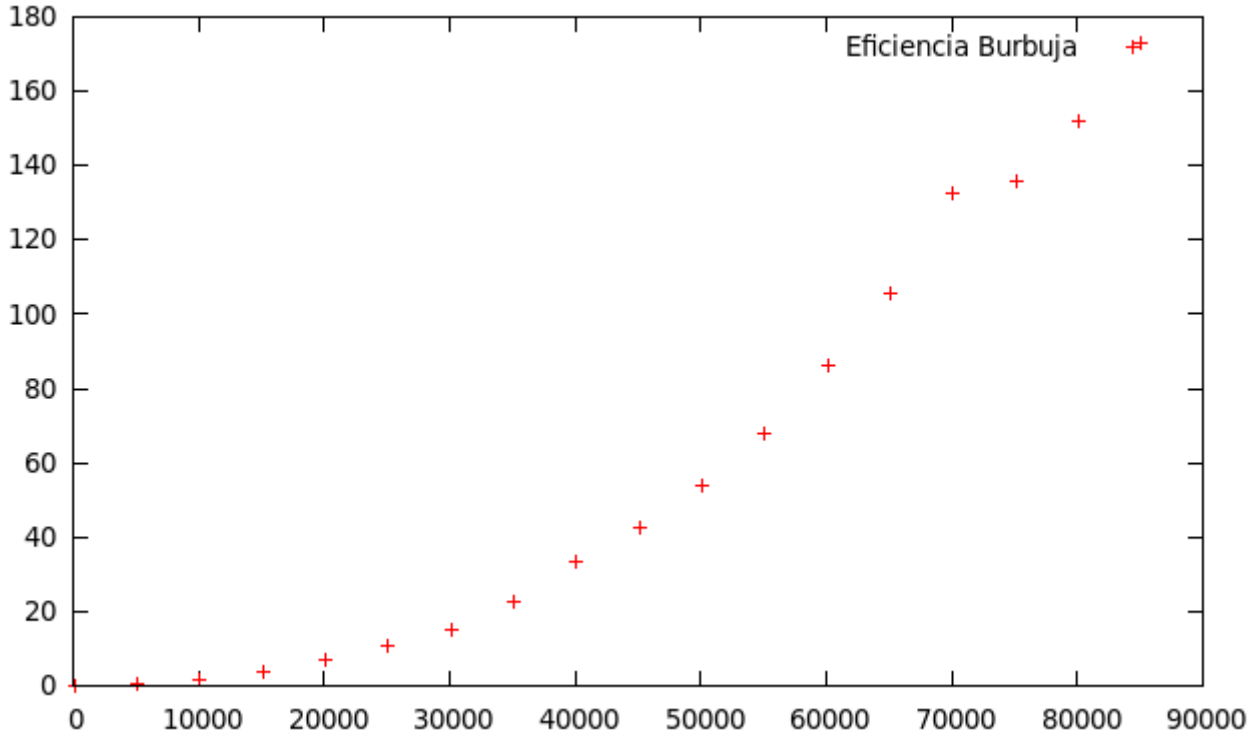
$-/burbuja > burbuja.dat

burbuja.dat

```
100 0
5100 0.41
10100 1.71
15100 3.9
20100 6.93
25100 10.69
30100 15.24
35100 22.85
40100 33.18
45100 42.59
50100 54.02
55100 67.71
60100 86.41
65100 105.6
70100 132.56
75100 135.94
80100 151.93
85100 172.92
```

Laura Tirado López
María del Mar García Cabello

$gnuplot
$gnuplot > plot 'burbuja.dat' title 'Eficiencia Burbuja'

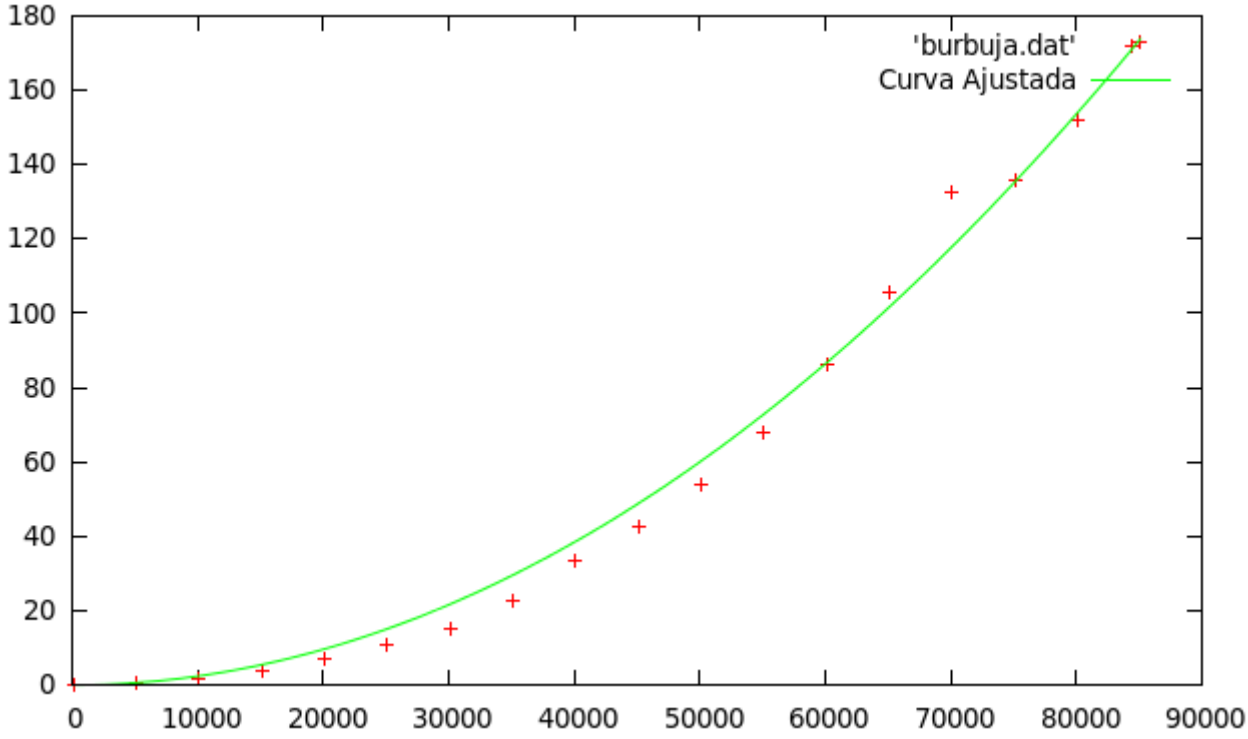


$gnuplot > f(x) = a*x*x
$gnuplot > fit f(x) 'burbuja.dat' via a

```
After 5 iterations the fit converged.
final sum of squares of residuals : 485.489
rel. change during last iteration : -3.51255e-16

degrees of freedom    (FIT_NDF)                        : 17
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 5.34399
variance of residuals (reduced chisquare) = WSSR/ndf   : 28.5582

Final set of parameters            Asymptotic Standard Error
=======================            ==========================

a               = 2.39992e-08      +/- 3.725e-10    (1.552%)

correlation matrix of the fit parameters:

               a
a               1.000
```

Laura Tirado López
María del Mar García Cabello

$gnuplot > plot 'burbuja.dat', f(x) title 'Curva ajustada'



Laura Tirado López
María del Mar García Cabello

-Entrada: quijote.txt

TERMINAL:

$g++ ordenacion.cpp -o ordenacion
$./ordenacion

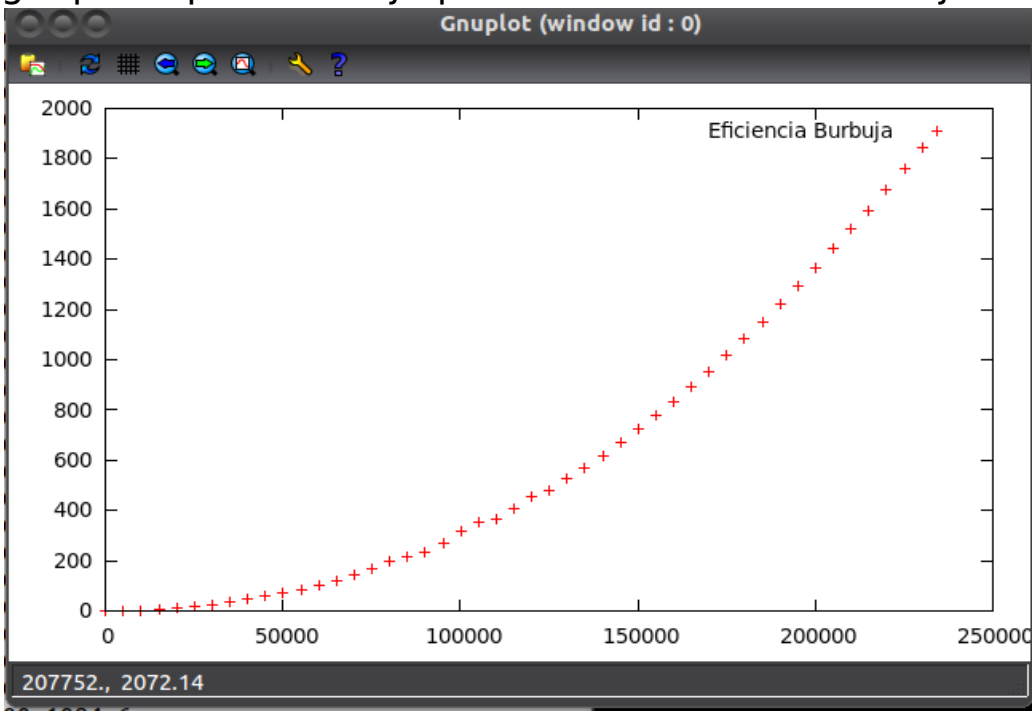
```
maria@maria-VirtualBox:~/ED$ g++ ordenacion.cpp -o ordenacion
maria@maria-VirtualBox:~/ED$ ./ordenacion
100 0.000471
5100 0.558371
10100 2.22368
15100 4.96392
20100 8.68005
25100 13.6949
30100 19.7626
35100 27.0435
40100 35.3855
45100 44.7081
50100 55.8237
55100 67.1486
60100 78.8723
65100 93.6849
70100 108.326
75100 124.177
80100 141.309
85100 159.762
```

$./ordenacion > burbuja.dat

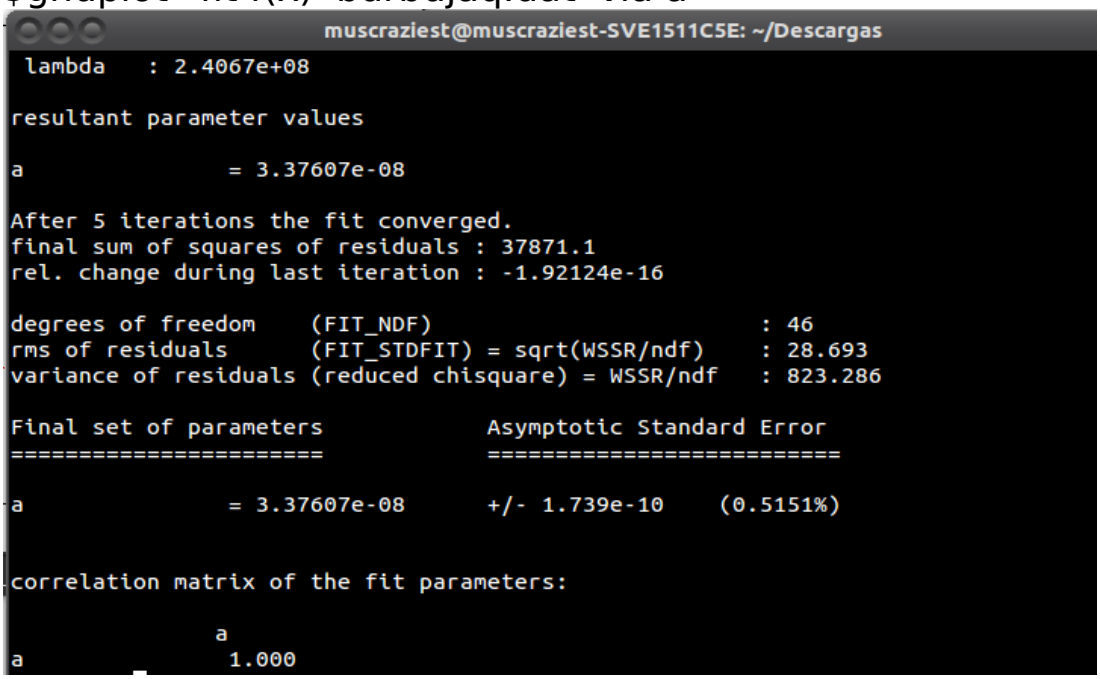
```
100 0.000554
5100 0.704364
10100 2.83827
15100 6.42529
20100 11.5231
25100 18.0471
30100 25.9723
35100 35.5312
40100 46.4673
45100 58.1908
50100 71.0478
55100 85.9506
60100 102.944
65100 120.429
70100 145.015
75100 166.225
80100 197.815
85100 218.494
90100 236.443
95100 271.906
```

Laura Tirado López
María del Mar García Cabello

$gnuplot
$gnuplot> plot 'burbujaq.dat' title 'Eficiencia Burbuja'

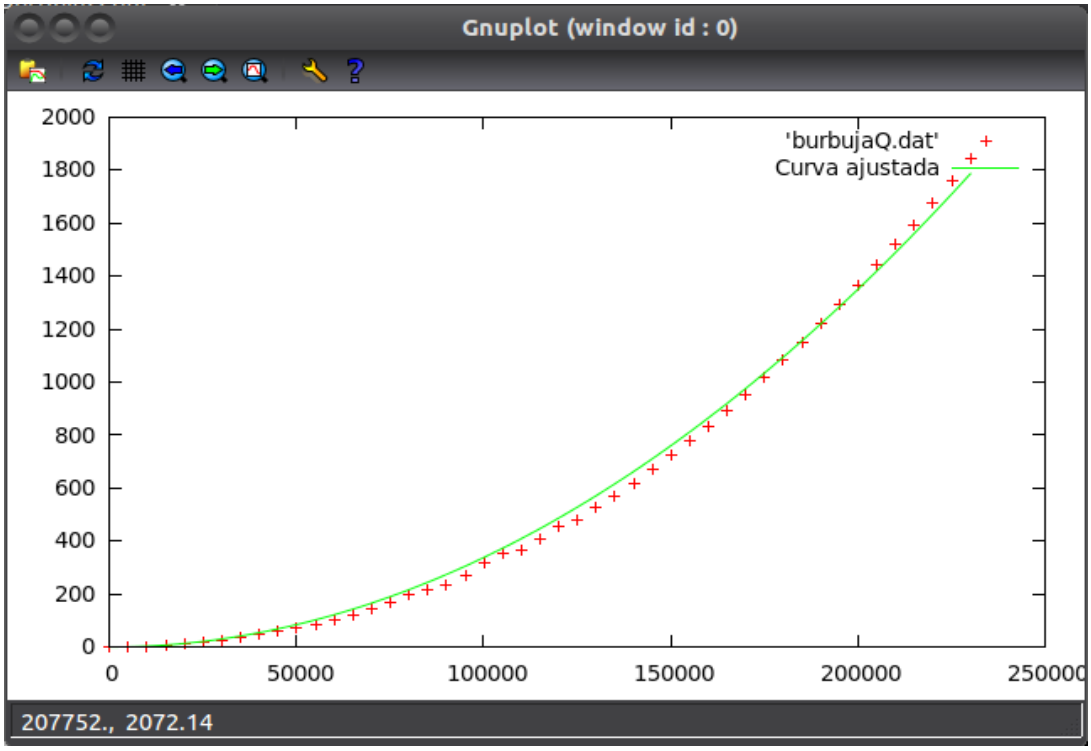


$gnuplot>f(x)=a*x*x
$gnuplot>fit f(x) 'burbujaq.dat' via a



Laura Tirado López
María del Mar García Cabello

$gnuplot>plot 'burbujaq.dat', f(x) title 'Curva ajustada'



b) Inserción

Nuestro algoritmo de insercción:

```
void insercion(vector<string> & T, int inicial, int final) {

        int i, j;
        string aux;

        for (i = inicial+1; i < final; i++){

                aux = T[i];
                j = i-1;

                while( (j >= inicial) && (T[j] > aux) ){

                        T[j+1] = T[j];
                        j--;
                }

                T[j+1] = aux;
        }
}
```

Laura Tirado López
María del Mar García Cabello

Práctica 1: Eficiencia

b.1) Análisis de eficiencia teórico

$$\sum_{i=inicio+1}^{final}\sum_{inicio}^{j=i-1} a=\sum_{i=1}^{n}\sum_{0}^{i-1} a=\sum_{i=1}^{n} a(i-1)=a\sum_{i=1}^{n} i-\sum_{i=1}^{n} 1=\frac{an^2-3an+2a}{2}=\frac{a}{2}n^2-\frac{3a}{2}n+a$$

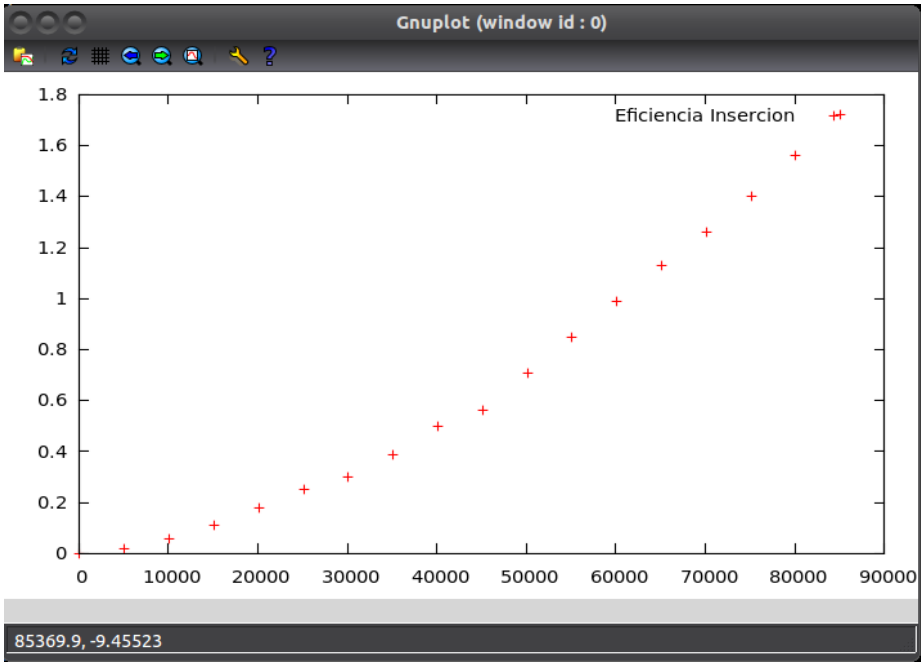
El algoritmo es de orden O(n²).

b.2) Análisis de eficiencia práctico

-Entrada: lema.txt

TERMINAL:

```
$g++ ordenacion.cpp -o ordenacion
$./ordenacion
$./ordenacion >> insercion.dat
$gnuplot
$gnuplot > plot 'insercion.dat '  title 'Eficiencia Inserción'
```

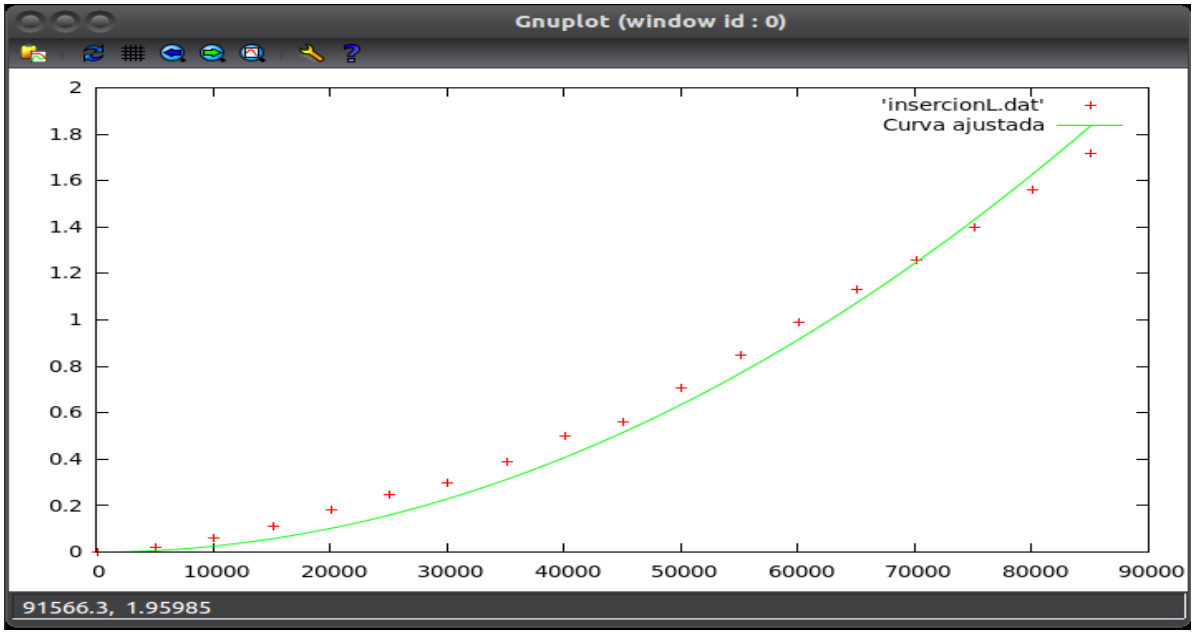


Laura Tirado López
María del Mar García Cabello

$gnuplot> f(x)=a*x*x
$gnuplot>fit f(x) 'insercion.dat' via a

```
a               = 4.16206e-10
/

 Iteration 4
 WSSR        : 0.0791459         delta(WSSR)/WSSR   : -68.5311
 delta(WSSR) : -5.42395          limit for stopping : 1e-05
 lambda      : 338115

resultant parameter values

a               = 2.53854e-10
/

 Iteration 5
 WSSR        : 0.0791459         delta(WSSR)/WSSR   : -2.11991e-13
 delta(WSSR) : -1.67782e-14      limit for stopping : 1e-05
 lambda      : 33811.5

resultant parameter values

a               = 2.53854e-10

After 5 iterations the fit converged.
final sum of squares of residuals : 0.0791459
rel. change during last iteration : -2.11991e-13

degrees of freedom    (FIT_NDF)                        : 17
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 0.0682323
variance of residuals (reduced chisquare) = WSSR/ndf   : 0.00465564

Final set of parameters            Asymptotic Standard Error
=======================            ==========================

a               = 2.53854e-10      +/- 4.757e-12    (1.874%)

correlation matrix of the fit parameters:

                a
a               1.000
```

$gnuplot>plot 'insercion.dat', f(x) title 'Curva ajustada'



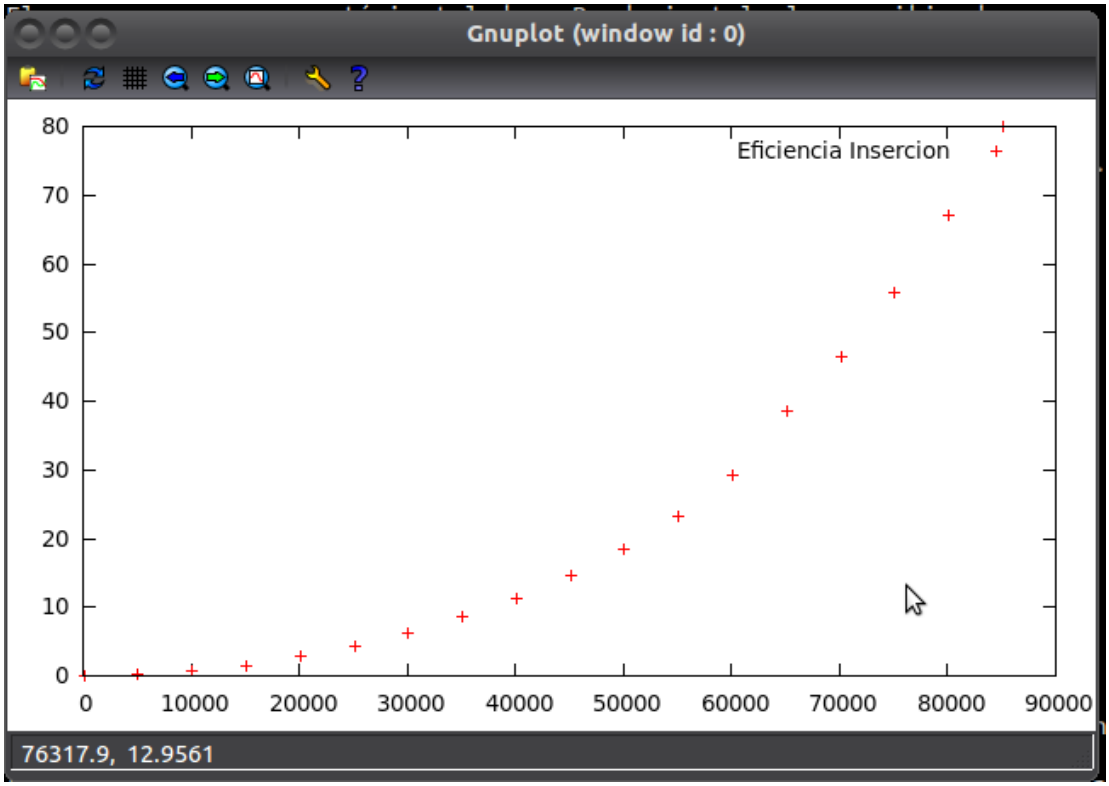
Laura Tirado López
María del Mar García Cabello

Práctica 1: Eficiencia

-Entrada: quijote.txt

TERMINAL:

```
$g++ ordenacion.cpp -o ordenacion
$./ordenacion
$./ordenacion >> insercion.dat
$gnuplot
$gnuplot > plot 'insercion.dat '  title 'Eficiencia Inserción'
```

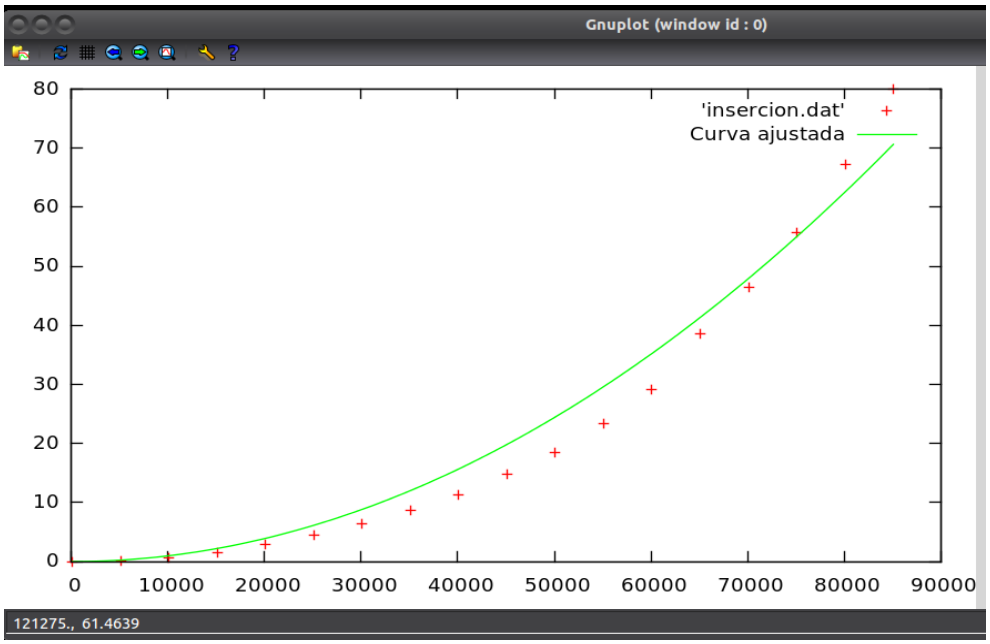


Laura Tirado López
María del Mar García Cabello

$gnuplot> f(x)=a*x*x
$gnuplot>fit f(x) 'insercion.dat' via a

```
a               = 9.91697e-09
/

 Iteration 4
 WSSR        : 297.457           delta(WSSR)/WSSR   : -0.0182344
 delta(WSSR) : -5.42395          limit for stopping : 1e-05
 lambda      : 338115

resultant parameter values

a               = 9.75462e-09
****/

 Iteration 5
 WSSR        : 297.457           delta(WSSR)/WSSR   : -3.82196e-16
 delta(WSSR) : -1.13687e-13      limit for stopping : 1e-05
 lambda      : 3.38115e+08

resultant parameter values

a               = 9.75462e-09

After 5 iterations the fit converged.
final sum of squares of residuals : 297.457
rel. change during last iteration : -3.82196e-16

degrees of freedom    (FIT_NDF)                        : 17
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 4.183
variance of residuals (reduced chisquare) = WSSR/ndf   : 17.4975

Final set of parameters            Asymptotic Standard Error
=======================            ==========================

a               = 9.75462e-09      +/- 2.916e-10    (2.989%)

correlation matrix of the fit parameters:

                a
a               1.000
```

$gnuplot>plot 'insercion.dat', f(x) title 'Curva ajustada'



Laura Tirado López
María del Mar García Cabello

c) Selección

Nuestro método de selección:

*ordenacion.cpp

```
int PosMenor(vector<string> & T, int final, int inicio){

        int pos;
        int i=inicio;
        string menor;

        menor = T[inicio];
        pos = inicio;

        for(int i=inicio+1; i < final; i++){

                if(menor < T[i]){

                        menor = T[i];
                        pos = i;
                }
        }

        return pos;
}

void seleccion(vector<string> & T, int inicial, int final) {

        int pos_men;
        string temp;

        for (int i=inicial; i<final - 1; i++) {
          pos_men = PosMenor(T, final, i);
          temp = T[i];
          T[i] = T[pos_men];
          T[pos_men] = temp;
        }

}
```

c.1) Análisis de eficiencia teórico

$$\sum_{i=inicio}^{final-1}\sum_{j=i+1}^{final} a=\sum_{i=0}^{n-1}\sum_{j=i+1}^{n} a=\sum_{i=0}^{n-1} a(n-i-1)=a\sum_{i=0}^{n-1} n-\sum_{i=0}^{n-1} i-\sum_{i=0}^{n-1} 1=\frac{an^2-3an}{2}=\frac{a}{2}n^2-\frac{3a}{2}n$$

El algoritmo es de orden 0(n²).

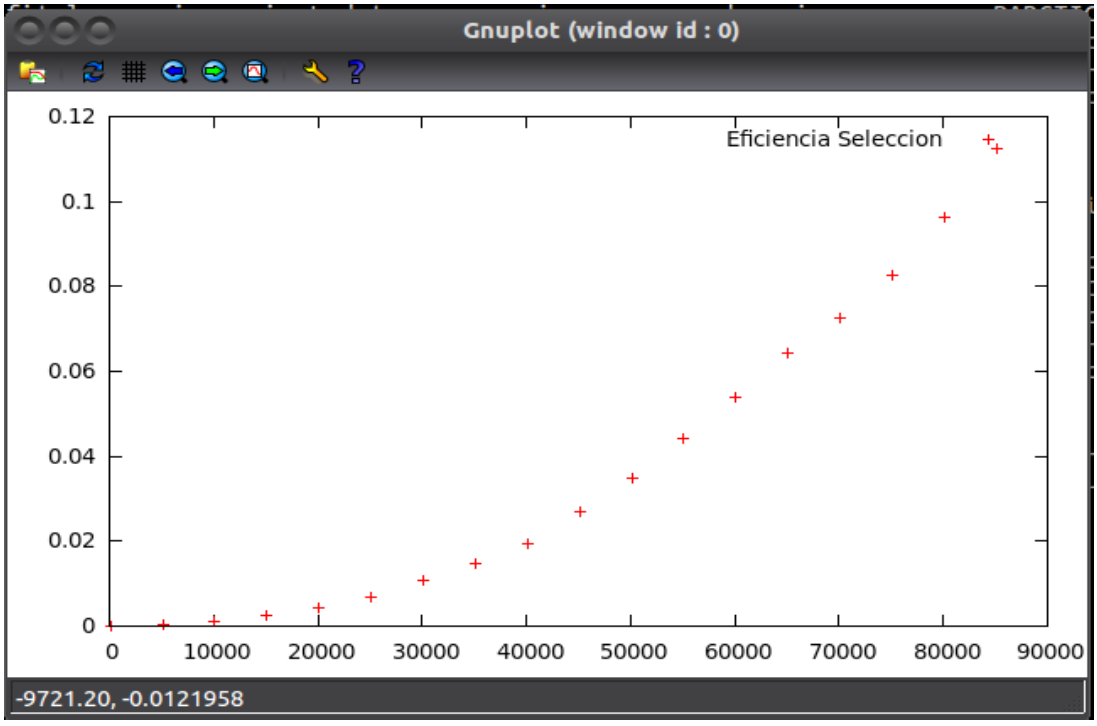
Laura Tirado López
María del Mar García Cabello

c.2) Análisis de eficiencia práctico

-Entrada: lema.txt

TERMINAL:

```
$ g++ ordenacion-sel.cpp -o seleccion
$./seleccion
$./seleccion > seleccionL.dat
$gnuplot
$gnuplot> plot 'seleccionL.dat' title 'Eficiencia Seleccion'
```



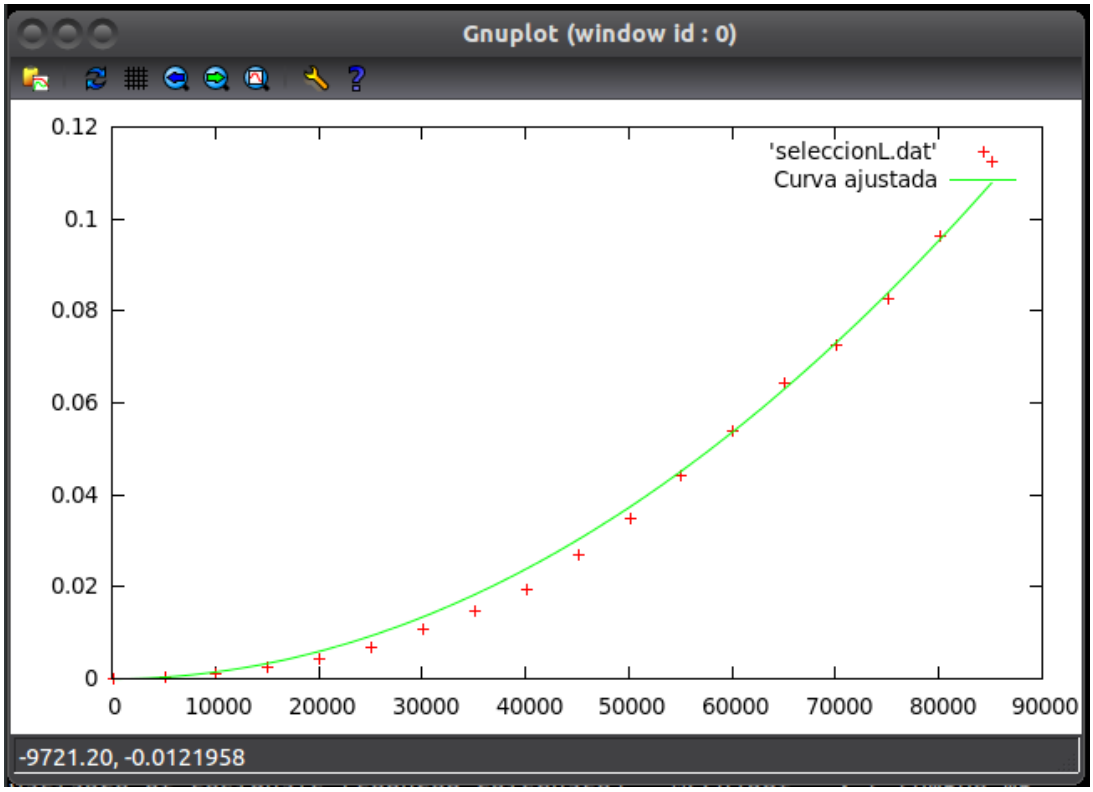
Laura Tirado López
María del Mar García Cabello

```
$gnuplot> f(x)=a*x*x
$gnuplot>fit f(x) 'seleccionL.dat' via a
```

```
a               = 1.77244e-10
/

 Iteration 4
 WSSR        : 9.50418e-05       delta(WSSR)/WSSR   : -57069.2
 delta(WSSR) : -5.42395          limit for stopping : 1e-05
 lambda   : 338115

resultant parameter values

a               = 1.48917e-11
/

 Iteration 5
 WSSR        : 9.50418e-05       delta(WSSR)/WSSR   : -1.76139e-10
 delta(WSSR) : -1.67405e-14      limit for stopping : 1e-05
 lambda   : 33811.5

resultant parameter values

a               = 1.48917e-11

After 5 iterations the fit converged.
final sum of squares of residuals : 9.50418e-05
rel. change during last iteration : -1.76139e-10

degrees of freedom    (FIT_NDF)                        : 17
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 0.00236446
variance of residuals (reduced chisquare) = WSSR/ndf   : 5.59069e-06

Final set of parameters            Asymptotic Standard Error
=======================            ==========================

a               = 1.48917e-11      +/- 1.648e-13    (1.107%)


correlation matrix of the fit parameters:

               a
a               1.000
```

Laura Tirado López
María del Mar García Cabello

$gnuplot>plot 'seleccionL.dat', f(x) title 'Curva ajustada'



-Entrada: quijote.txt

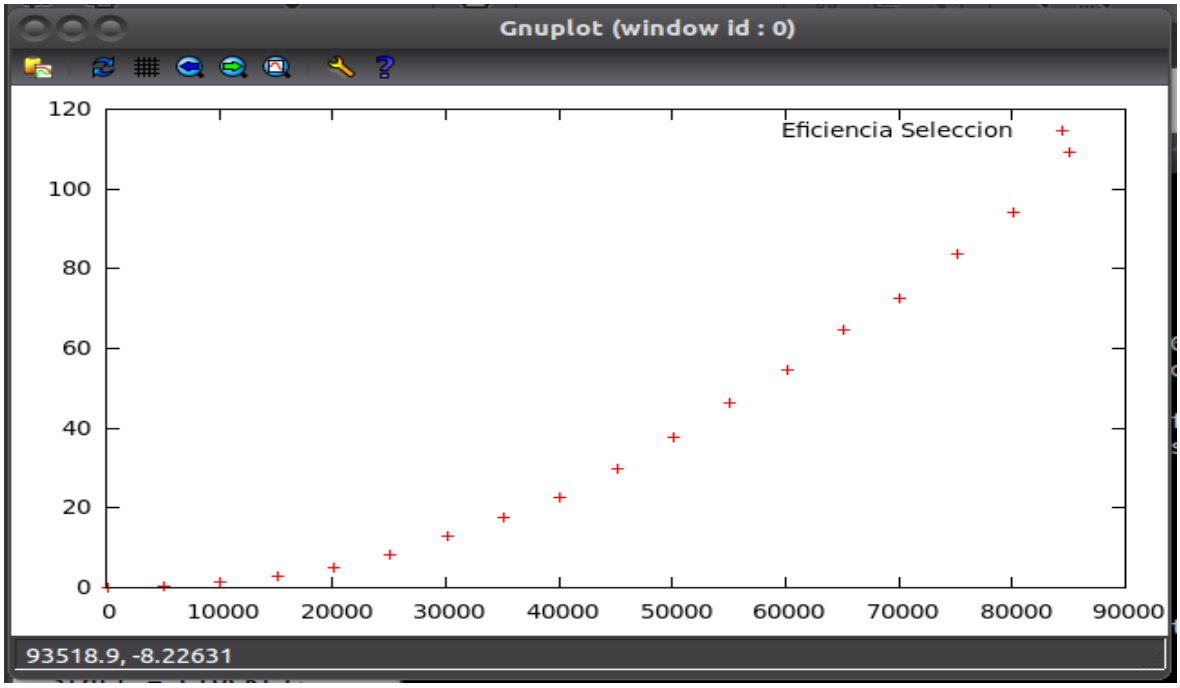
TERMINAL:

$ g++ ordenacion-sel.cpp -o seleccion
$./seleccion

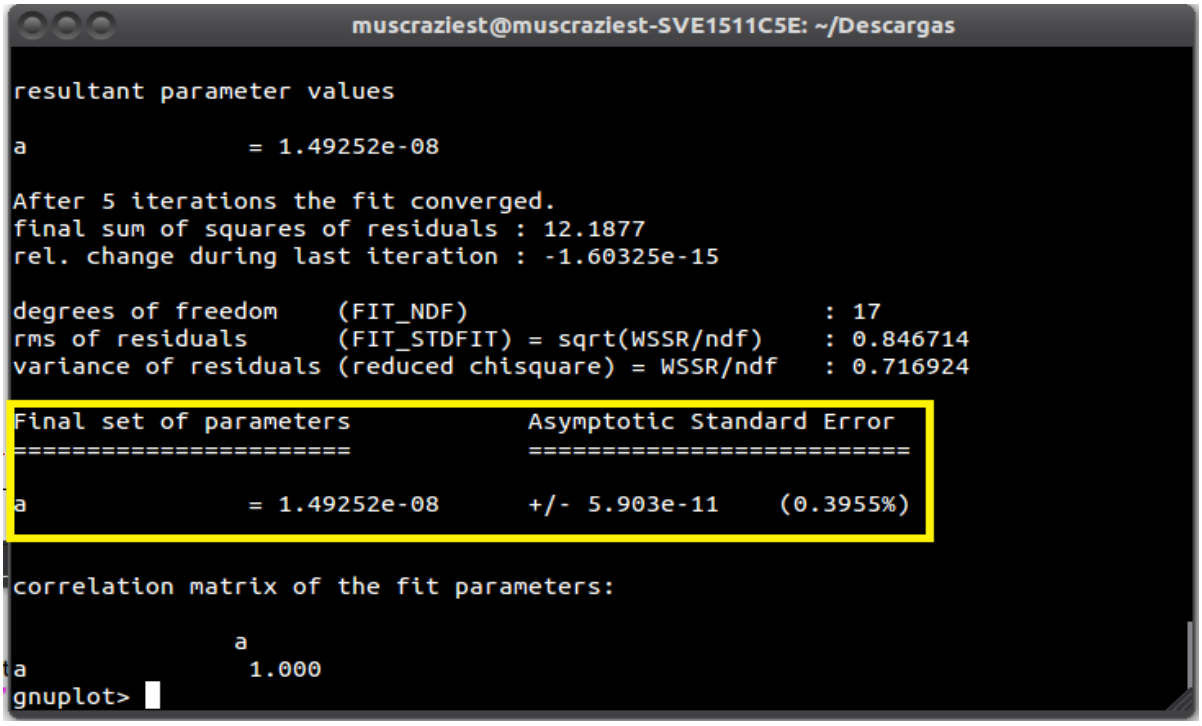
```
muscraziest@muscraziest-SVE1511C5E:~/Descargas$ g++ ordenacion.cpp -o seleccion
muscraziest@muscraziest-SVE1511C5E:~/Descargas$ ./seleccion
100     0
5100    0.32
10100    1.26
15100    2.88
20100    5.15
25100    8.08
30100    11.8
35100    17.34
40100    23.87
45100    32.09
50100    41.1
55100    51.72
60100    63.74
65100    77.02
70100    92.99
75100    110.13
80100    129.13
85100    148.93
90100    170.47
```

Laura Tirado López
María del Mar García Cabello

```
$./seleccion > seleccionL.dat
$gnuplot
$gnuplot> plot 'seleccionL.dat' title 'Eficiencia Seleccion'
```

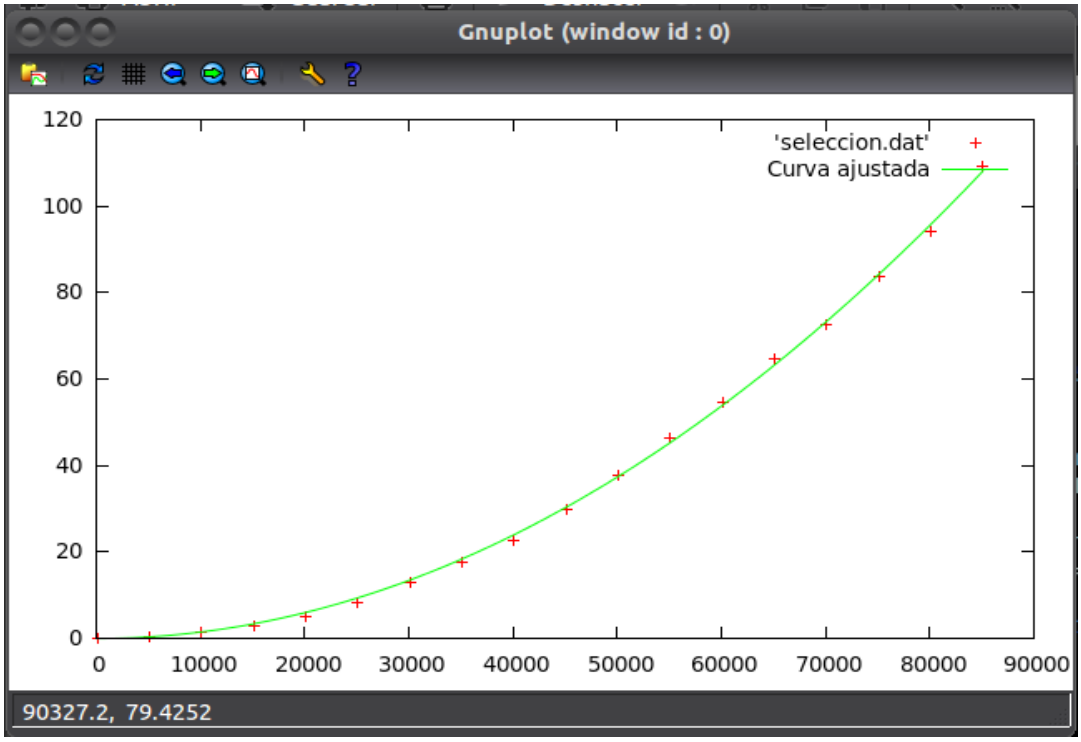



```
$gnuplot> f(x)=a*x*x
$gnuplot>fit f(x) 'seleccionL.dat' via a
```

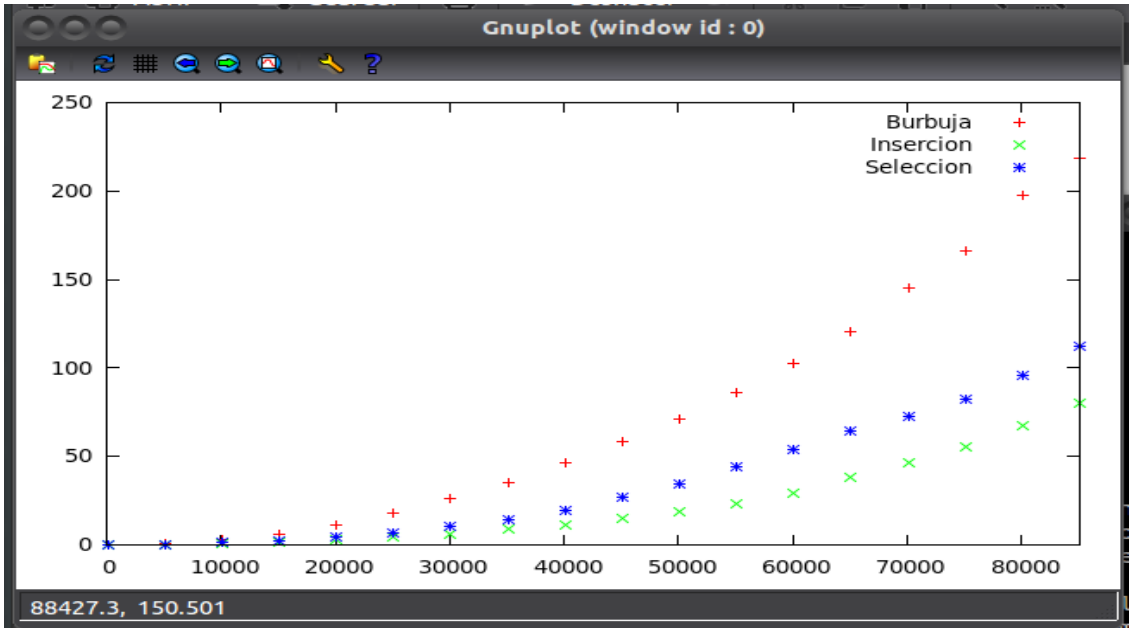


Laura Tirado López
María del Mar García Cabello

$gnuplot>plot 'seleccionL.dat', f(x) title 'Curva ajustada'



Ahora vamos a comparar los tres algoritmos:
Aunque los tres son de orden O(n²) no crecen de la misma manera:



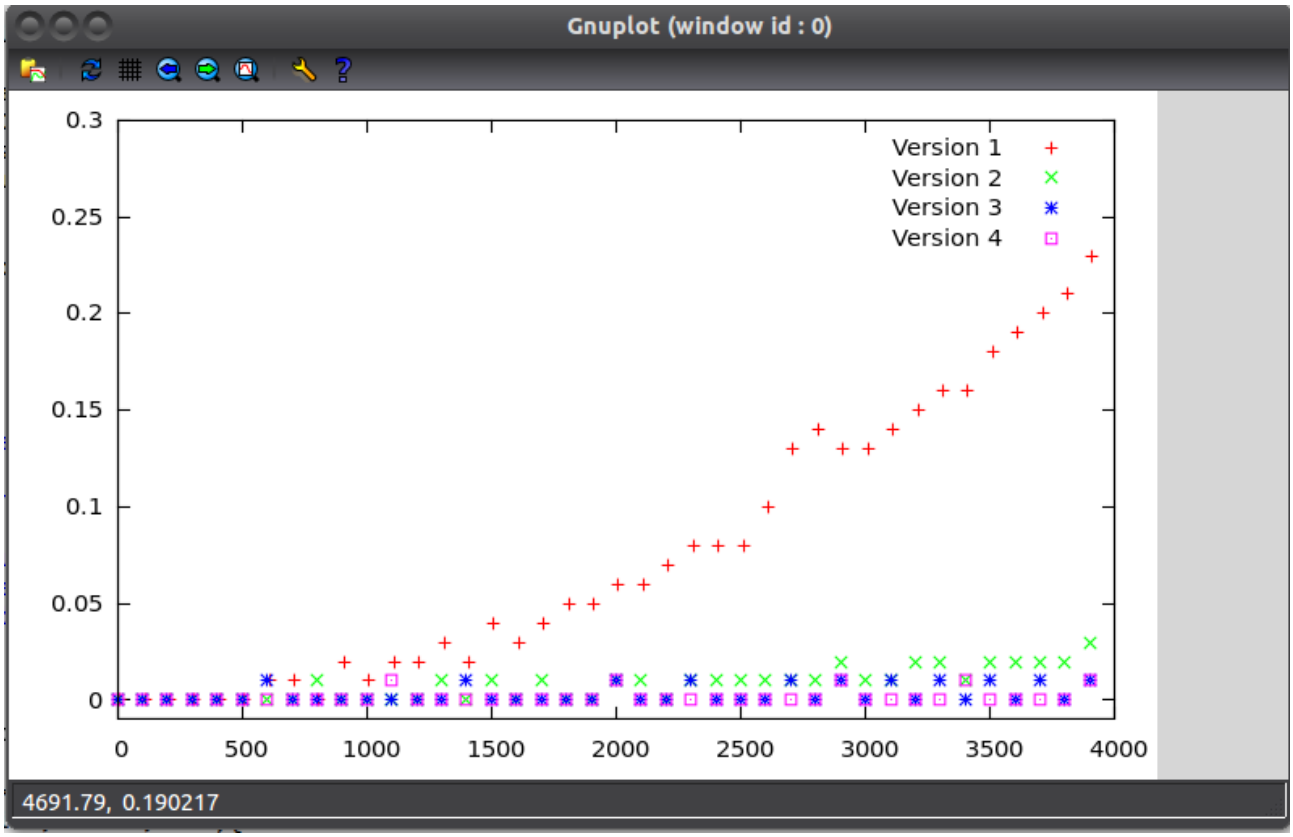
Laura Tirado López
María del Mar García Cabello

Como podemos ver, el algoritmo de inserción crece más lentamente que los otros dos, por lo que es mucho más eficiente que el de burbuja, y más eficiente que el de selección, sobretodo con grandes tamaños de entrada.

En pequeños tamaños de entrada, no se aprecian diferencias entre los tres algoritmos.

Laura Tirado López
María del Mar García Cabello

EJERCICIO 3



En las tres primeras versiones se usa como estructura de datos dos vectores de la STL, uno para guardar las palabras leídas y otro para guardar el número de repeticiones de cada palabra.
Sin embargo en la versión 4, la estructura que se utiliza es un map, un contenedor asociativo para guardar parejas de datos, en el que cada pareja está formada por una palabra y el valor de repeticiones de dicha palabra en el texto.

Es más eficiente el map, ya que se guarda tanto las palabras leídas como su número de repeticiones como un “único” dato del map, una pareja <string, int>.

Laura Tirado López
María del Mar García Cabello