UNIVERSIDAD DE GRANADA

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

---

ALGORÍTMICA
Práctica 1 : Eficiencia de algoritmos

Autores :
Elvira Castillo Fernández
David Gil Bautista
Jose Luis Izquierdo Mañas
Freddy A. Jaramillo López
Alejandro Jerónimo Fuentes
Gregorio Vidoy Fajardo

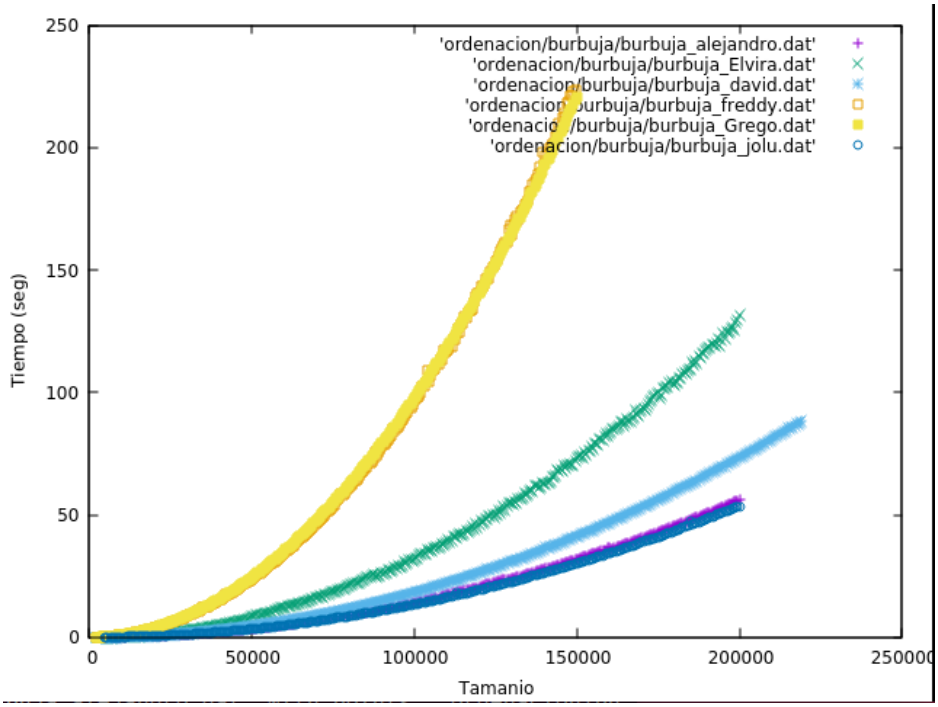
Fecha : 21 de marzo de 2017
Grupo de prácticas : C1
Subgrupo : 4

1. Calcule la eficiencia empírica, siguiendo las indicaciones de la sección 3. Defina adecuadamente los tamaños de entrada de forma tal que se generen al menos 25 datos. Incluya en la memoria tablas diferentes para los algoritmos de distinto orden de eficiencia (una con los algoritmos de orden O(n2), otra con los O(n log n), otra con O(n3) y otra con O(2n)).

Burbuja:

Hemos realizado mediciones de valores para el algoritmo de burbuja empezando en 2000 valores y terminando en ordenar vectores de 200000 valores, incrementando en cada medición el número de elementos en el vector de 500 en 500.

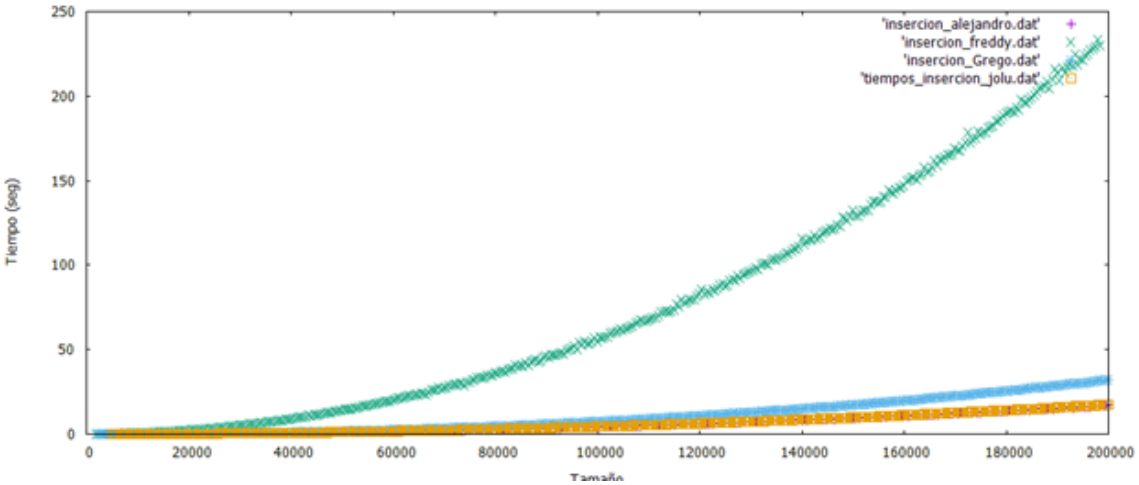
Hemos realizado estas mediciones en equipos con hardware diferente y con opciones distintas de compilación, y este ha sido el resultado:



Como podemos ver varían las gráficas dependiendo del hardware y de las opciones de compilación. Las de Grego y Fredi, están en equipos con similares características tanto hardware y con el mismo sistema operativo y sin ninguna optimización de compilación, las de Jolu, David y Alejandro están con optimización –O2 y la de Elvira está en otro hardware con otro sistema operativo diferente, esas son las diferencias que podemos ver en las ejecuciones.

Sólo añadiremos una parte de la tabla que se generó al tomar los valores para realizar la representación de la gráfica porque se tomaron muchísimas muestras y habría varias páginas con valores de tiempo y tamaño para cada uno de los algoritmos.(si se quiere ver los valores completos que se han tomado, están en la carpeta adjunta a este documento.)

TAMAÑOS	TIEMPOS(Seg)
2000	0.015625
2500	0.015625
3000	0.03125
3500	0.015625
4000	0.03125
4500	0.046875
5000	0.0625
5500	0.078125
6000	0.09375
6500	0.109375
…….	…….
143000	65.7188
143500	66.25
144000	68.6719
144500	68.125
145000	68.5
…….	…….
195500	122.25
196000	125.328
196500	124.375
197000	125.406
197500	126.344



Inserción

Se puede apreciar que los tiempos de ejecución son distintos según el ordenador que los ha ejecutado. Esto se debe a las diferencias de hardware de cada ordenador y también a la optimización a la hora de compilar el código. Los tiempos de ejecución de los archivos 'insercion_alejandro.dat' y 'tiempos_insercion_jolu.dat' han sido compilados con la opción -O2

Algunos de los valores que se han obtenido han sido:

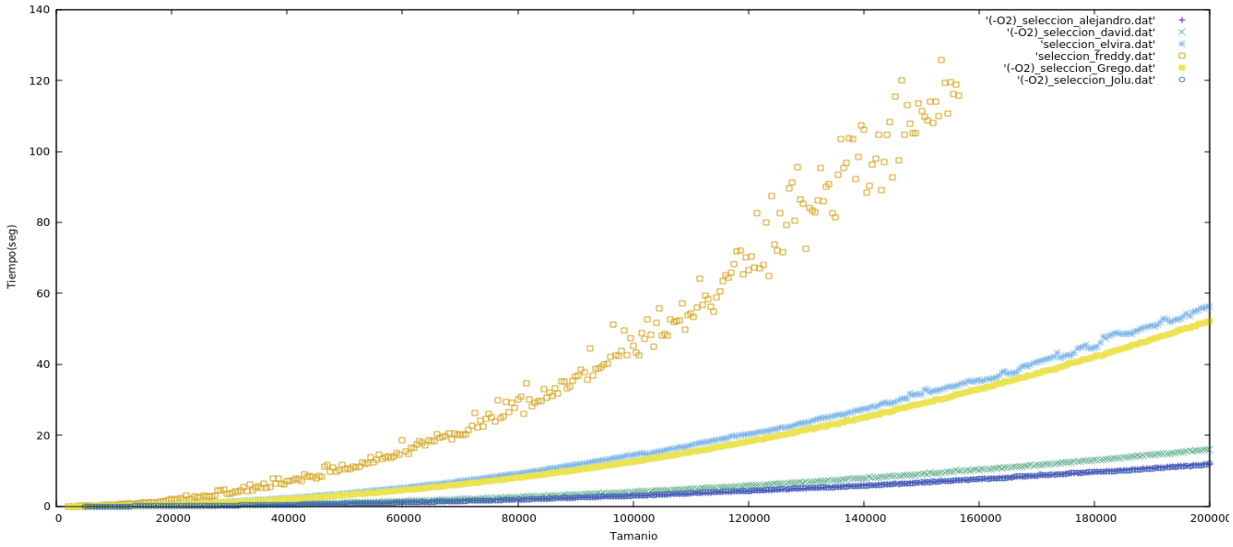
Tamaño de entrada	Tiempo (Seg)
6000	0.015625
8000	0.03125
9500	0.046875

107500	4.85938
108000	4.90625
108500	4.95312
199000	16.8906
199500	16.7344
200000	17.1719

Selección:

Hemos realizado mediciones de valores para el algoritmo de burbuja empezando en 2000 valores y terminando en ordenar vectores de 200000 valores, incrementando en cada medición el número de elementos en el vector de 500 en 500.

Hemos realizado estas mediciones en equipos con hardware diferente y con opciones distintas de compilación, y este ha sido el resultado:



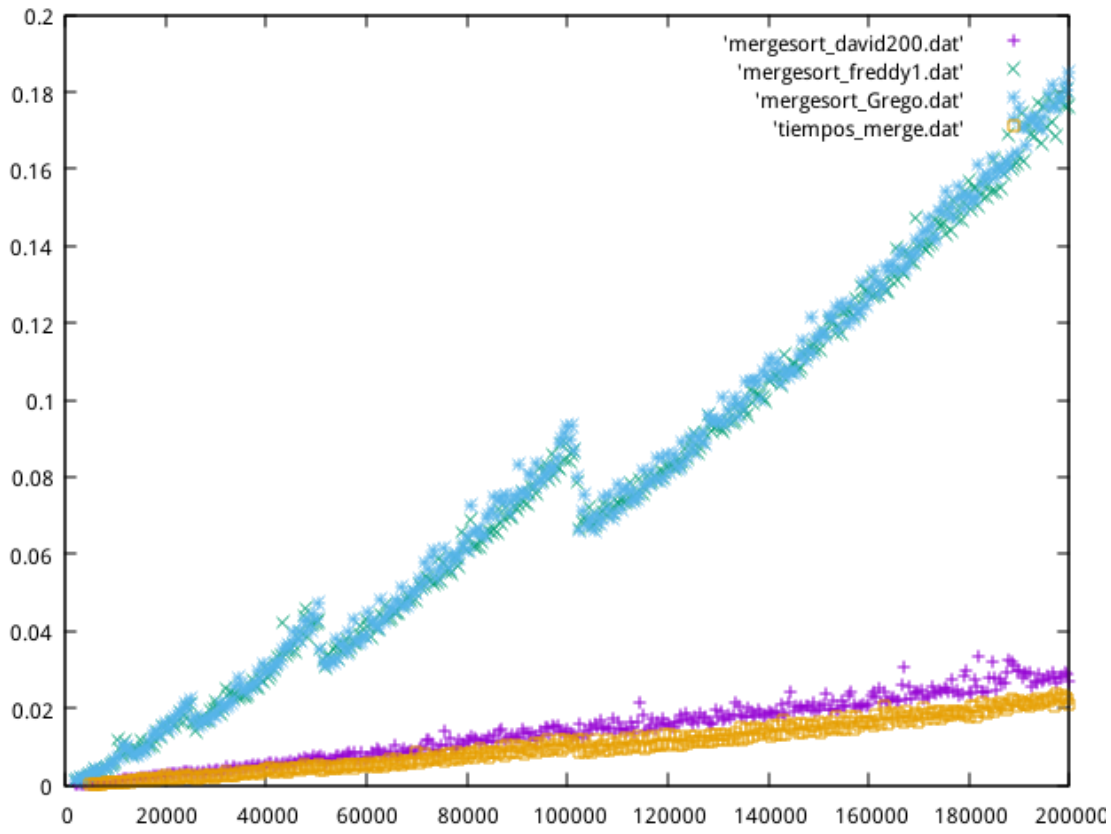
Se puede apreciar que los tiempos de ejecución son distintos según el ordenador que los ha ejecutado. Esto se debe a las diferencias de hardware de cada ordenador y también a la optimización a la hora de compilar el código. Los tiempos de ejecución de los archivos 'seleccion_elvira.dat' y 'seleccion_freddy.dat' han sido compilados sin optimización de compilación.

Algunos de los valores que se han obtenido han sido:

TAMAÑOS	TIEMPOS(Seg)
2000	0.00568777
2500	0.00804
3000	0.012405
3500	0.017437
4000	0.022724
4500	0.030412

5000	0.040013
5500	0.048152
6000	0.057192
6500	0.067044
…….	…….
143000	26.3186
143500	26.461
144000	26.6193
144500	26.6647
145000	26.8303
…….	…….
195500	50.1477
196000	50.2897
196500	50.3756
197000	50.7796
197500	50.8194

Mergesort:



En la gráfica anterior se muestran las ejecuciones de todos los compañeros del algoritmo *mergesort* para el mismo número de entradas en distintas máquinas.

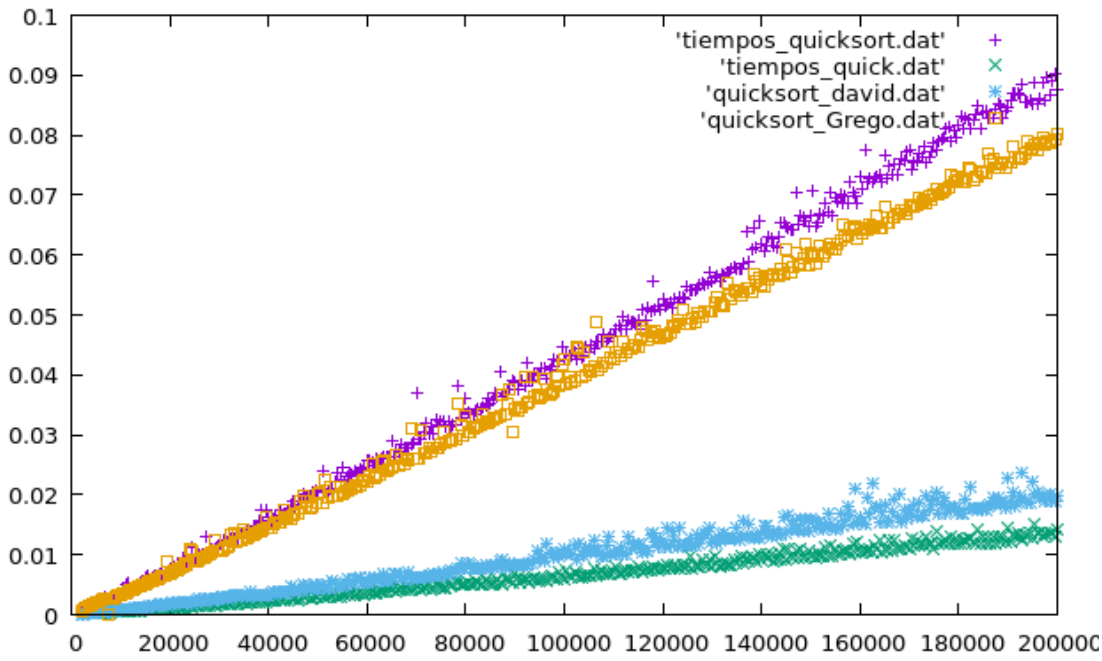
Podemos observar que hay dos nubes de puntos que se disparan y son notablemente superiores a las demás, esto se debe a que el algoritmo ha sido compilado sin usar optimización. Para las otras funciones se puede apreciar que los pequeños saltos que tiene este algoritmo son casi imperceptibles.

En la siguiente tabla podemos apreciar algunos de los valores que se han usado para representar una función de la gráfica anterior.

TAMAÑOS	TIEMPOS(Seg)
2000	0.00015162
2500	0.000198092
3000	0.000254718
3500	0.0002798
4000	0.000338138
4500	0.000375973
5000	0.00043228
5500	0.000499076
6000	0.000557404
6500	0.000545124
…….	…….
143000	0.01878
143500	0.018007
144000	0.021501
144500	0.024394
145000	0.01901
…….	…….
195500	0.028371
196000	0.02726
196500	0.029192
197000	0.028016
197500	0.026633

Como se puede observar los tiempos son mucho menores que para los algoritmos de la familia cuadrática.

Quicksort:



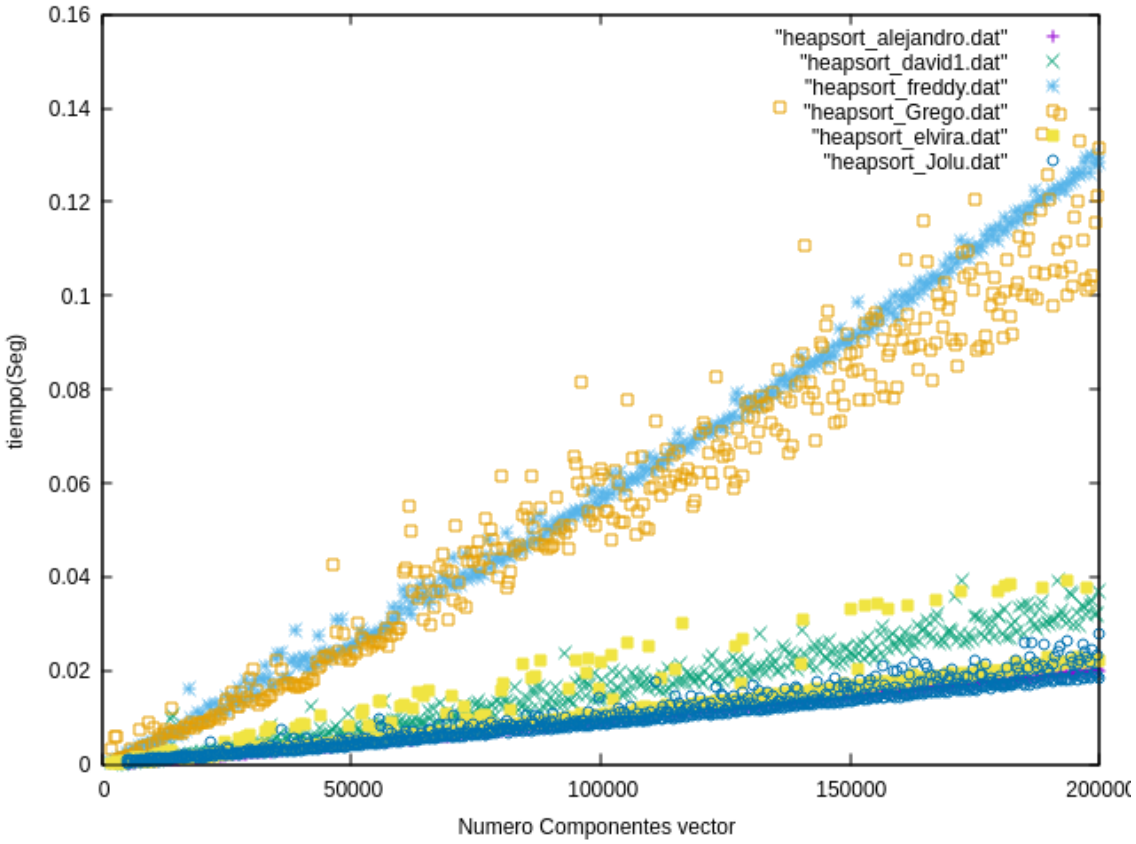
En la gráfica anterior se muestran las ejecuciones de todos los compañeros del algoritmo *quicksort* para el mismo número de entradas en distintas máquinas.

Podemos observar que hay dos nubes de puntos que se disparan y son notablemente superiores a las demás, esto se debe a que el algoritmo ha sido compilado sin usar optimización.

En la siguiente tabla podemos apreciar algunos de los valores que se han usado para representar una función de la gráfica anterior.

TAMAÑOS	TIEMPOS(Seg)
2000	0.00061
2500	0.000841
3000	0.00122
3500	0.001155
4000	0.001414
4500	0.001469
5000	0.0017
5500	0.002387
6000	0.002626
6500	0.002188
…….	…….
143000	0.061186
143500	0.063106
144000	0.065304
144500	0.065329
145000	0.064914
…….	…….
195500	0.088808
196000	0.085373
196500	0.086059
197000	0.08882
197500	0.085897

Heapsort:

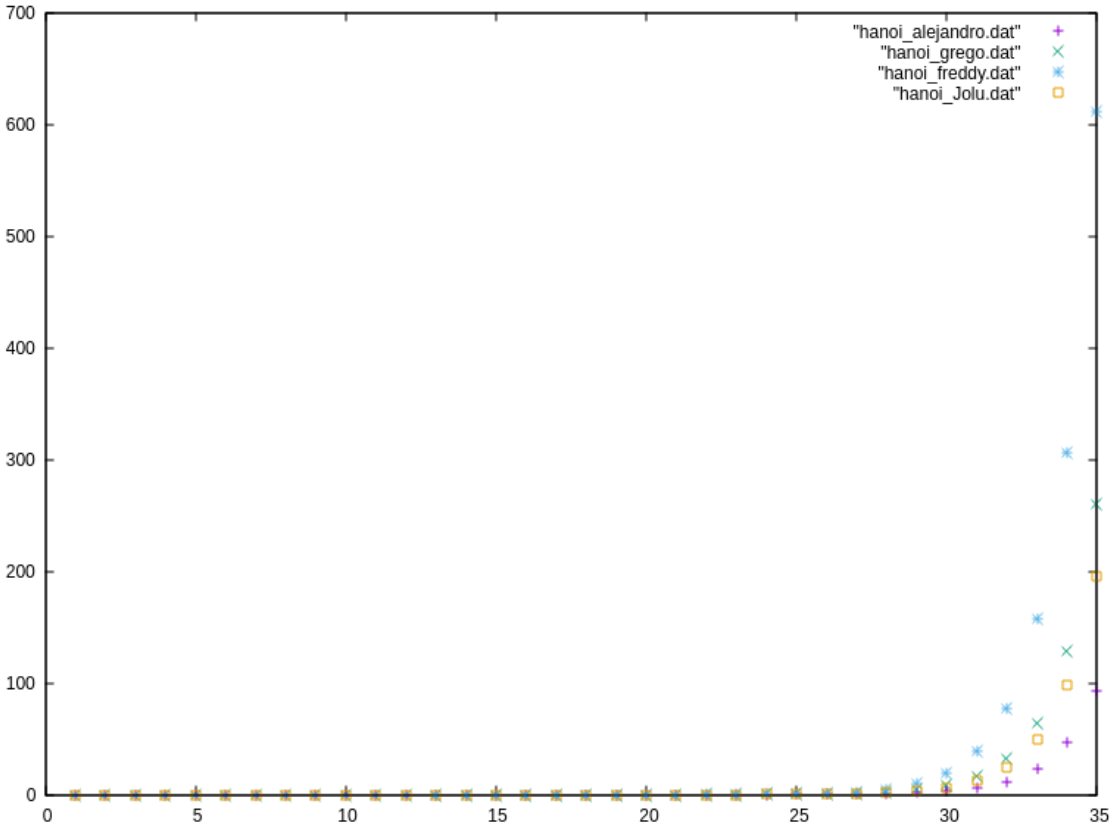


En la gráfica anterior se muestran las ejecuciones de todos los compañeros del algoritmo *quicksort* para el mismo número de entradas en distintas máquinas.

Los tiempos de ejecución de este algoritmo se pueden consultar en los archivos adjuntos.

Hanoi

En la gráfica anterior se muestran las ejecuciones de todos los compañeros del algoritmo de

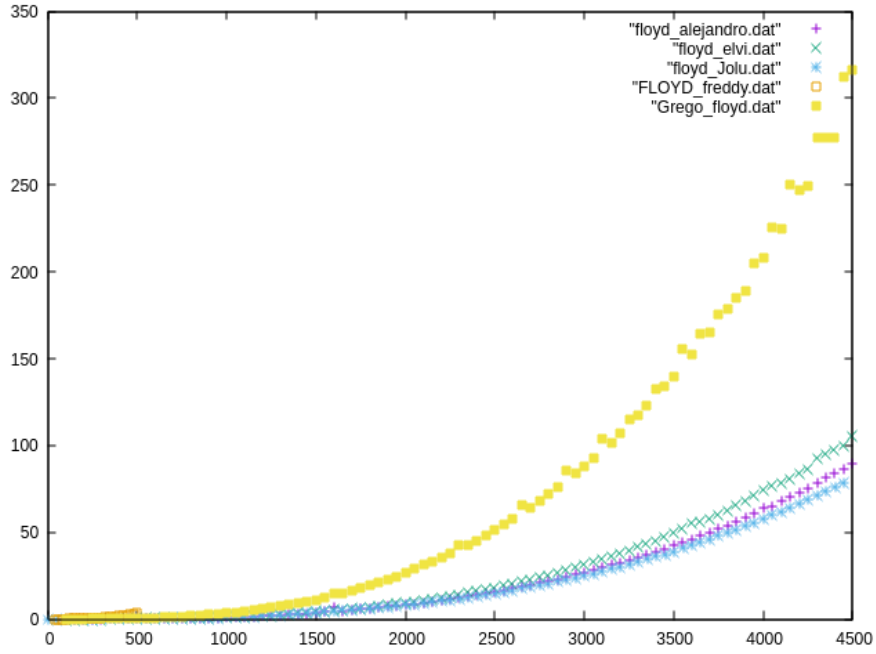


Hanoi para el mismo número de entradas en distintas máquinas.

Podemos observar el comportamiento puramente exponencial del algoritmo de las torres de Hanoi donde apreciamos que para entradas de hasta 30 discos se mantiene en tiempos estables y similares y a partir de ese umbral se dispara de una manera desmesurada.

Los tiempos de ejecución de este algoritmo se pueden consultar en los archivos adjuntos.

Floyd:



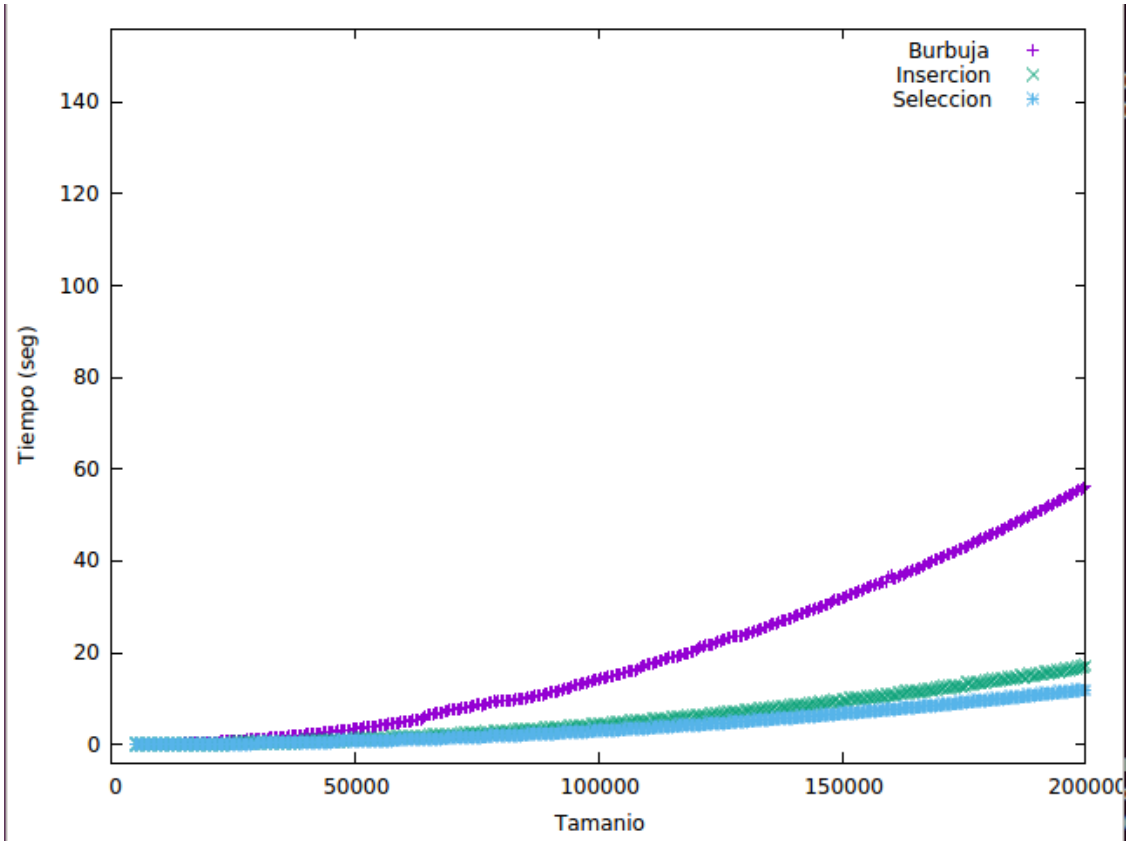
En la gráfica anterior

se muestran las ejecuciones de todos los compañeros del algoritmo de *Floyd* para el mismo número de entradas en distintas computadoras.

Los tiempos de ejecución de este algoritmo se pueden consultar en los archivos adjuntos 'floyd_nombre.dat'

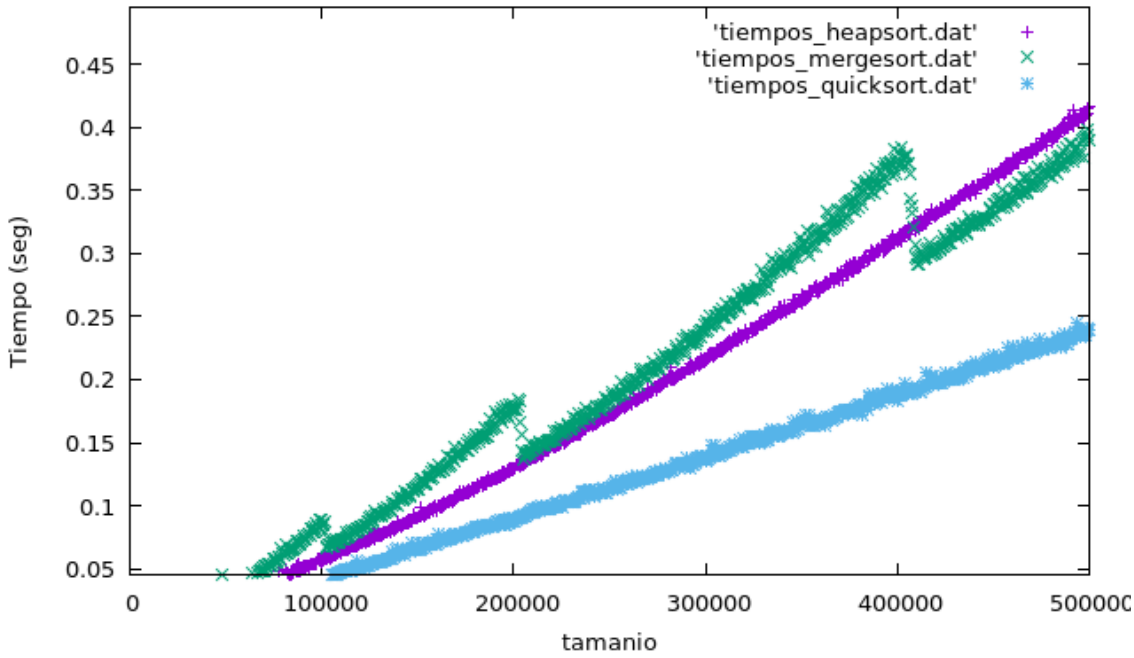
2. Con cada una de las tablas anteriores, genere un gráfico comparando los tiempos de los algoritmos. Indique claramente el significado de cada serie. Para los algoritmos que realizan la misma tarea (los de ordenación), incluya también una gráfica con todos ellos, para poder apreciar las diferencias en rendimiento de algoritmos con diferente orden de eficiencia.

Familia cuadrática:



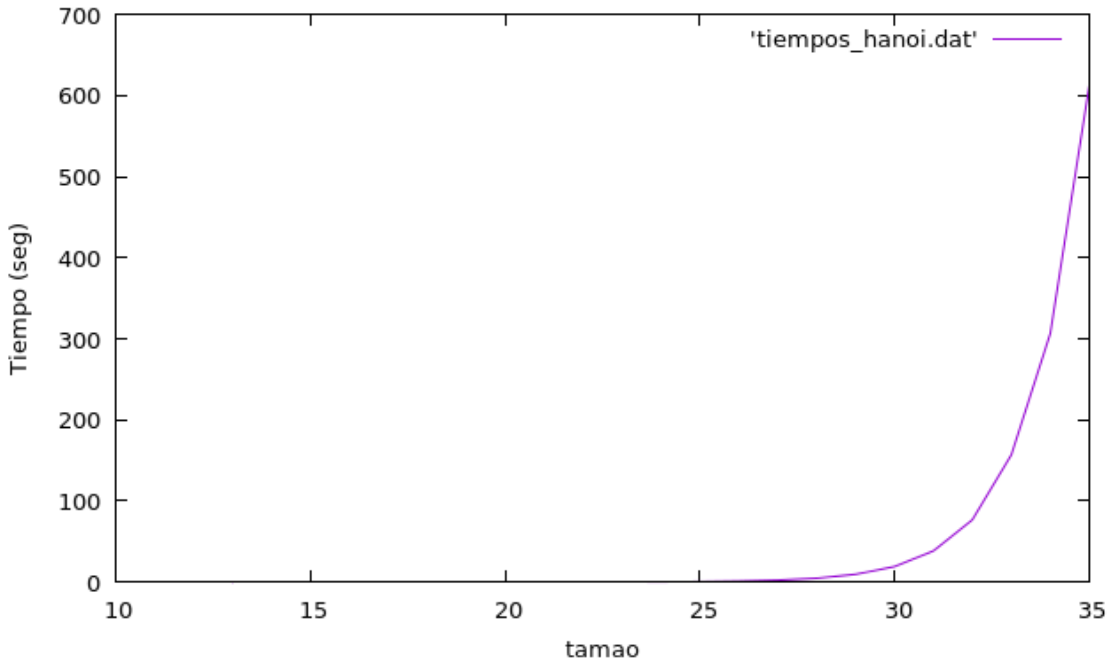
Como podemos ver en la gráfica, de la familia n^2 el que peores tiempos tiene, en el peor de los casos, para tamaños muy grandes de 'n' es el algoritmo burbuja, seguido del inserción.

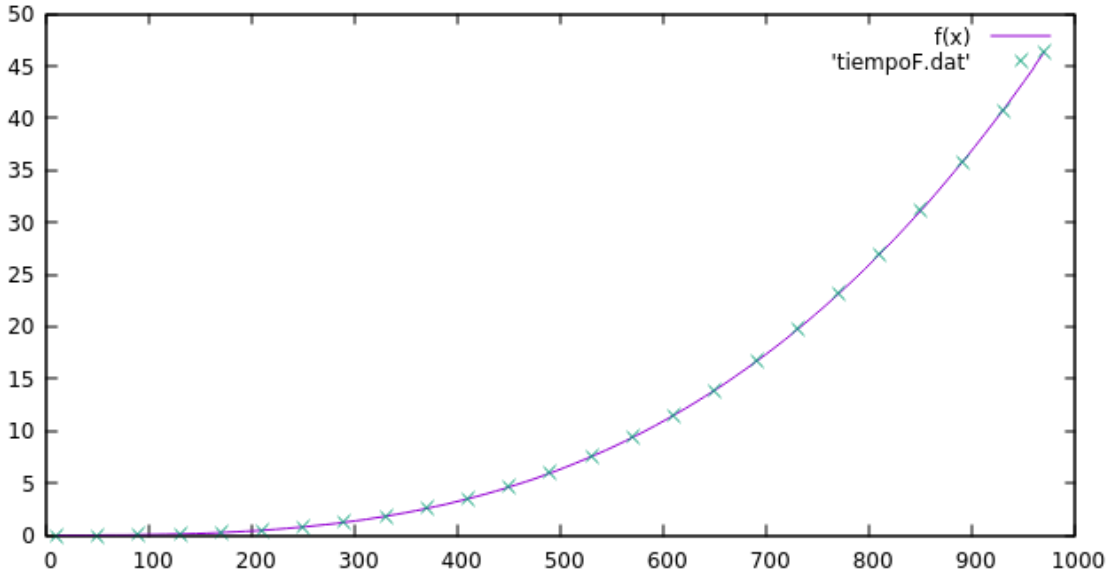
Familia logarítmica:



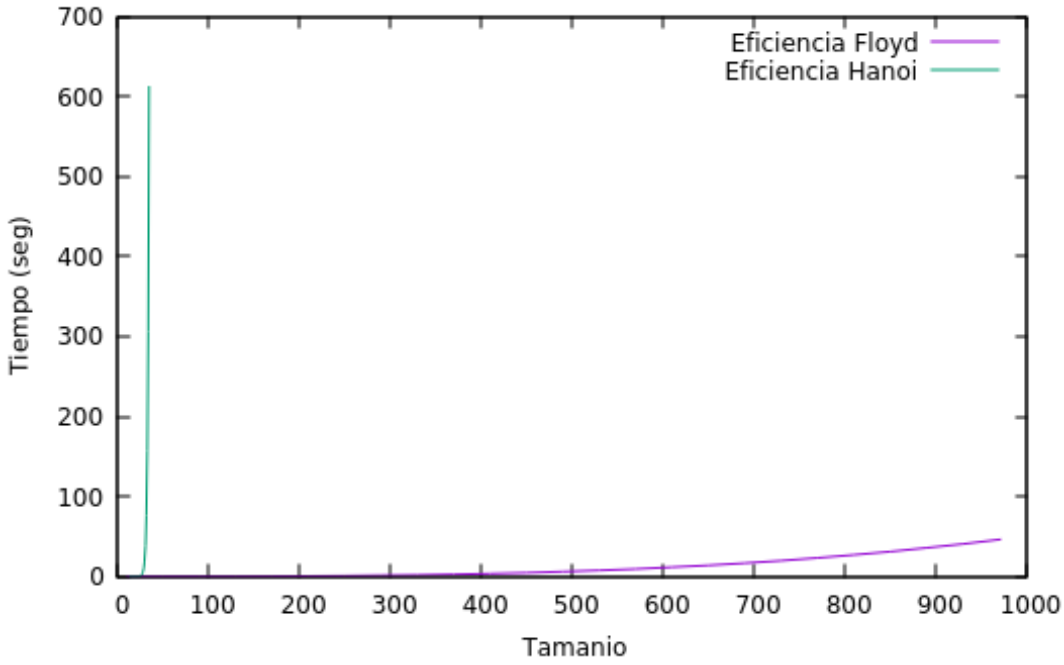
Al comparar los algoritmos de la familia logarítmica podemos observar que para valores pequeños el algoritmo mergesort es peor que los otros dos a comparar y que a partir de cierto punto para unos valores es ligeramente mejor que el heapsort. El mejor de todos es el algoritmo quicksort ya que a pesar de que se base en elegir un pivote e ir ordenando los valores mayores o menores que ese pivote al dividir el trabajo se recorren muchas menos posiciones y el ordenamiento es casi inmediato.

Familia cúbica y exponencial:





Funciones juntas:



Podemos observar que estos algoritmos poseen unos tiempos de ejecución bastante altos para la cantidad de entradas con las que estamos trabajando, es por lo que problemas como las Torres de Hanoi o encontrar el camino mínimo en un grafo son tareas con un alto tiempo computacional las cuales con las prestaciones que hoy en día existen siguen suponiendo un esfuerzo enorme.

3. Calcule también la eficiencia híbrida de todos los algoritmos, siguiendo las pautas indicadas en la sección 4. Pruebe también con otros ajustes que no se correspondan con la eficiencia teórica (ajuste lineal, cuadrático, etc) y compruebe la variación en la calidad del ajuste.

Burbuja:

En este caso hemos calculado la eficiencia híbrida del algoritmo burbuja, hemos hecho un ajuste a una función cuadrática y hemos calculado los coeficientes de la recta de regresión que más se adapta entre:

*F(x) = A0*x^2+A1*x+A2*

y la función que aparece al hacer la gráfica con los tiempos medidos por nosotros en uno de los ordenadores.

```
gnuplot> fit  a0*x*x+a1*x+a2 'ordenacion/burbuja/burbuja_Elvira.dat' via a0,a1,a2
iter      chisq       delta/lim  lambda   a0             a1             a2
   0 1.2880294052e+23   0.00e+00  1.04e+10    1.000000e+00   1.000000e+00   1.000000e+00
   1 9.0651473940e+16  -1.42e+11  1.04e+09    8.326924e-04   9.999938e-01   1.000000e+00
   2 3.3459170138e+11  -2.71e+10  1.04e+08   -6.231928e-06   9.999934e-01   1.000000e+00
   3 3.3456460907e+11  -8.10e+00  1.04e+07   -6.238779e-06   9.999625e-01   1.000000e+00
   4 3.3250403889e+11  -6.20e+02  1.04e+06   -6.219527e-06   9.968783e-01   9.999999e-01
   5 1.9393980371e+11  -7.14e+04  1.04e+05   -4.749205e-06   7.613334e-01   9.999952e-01
   6 1.9013316617e+08  -1.02e+08  1.04e+04   -1.455015e-07   2.382155e-02   9.999805e-01
   7 2.3385688299e+02  -8.13e+10  1.04e+03    3.255597e-09  -9.293010e-06   9.999796e-01
   8 2.1400187281e+02  -9.28e+03  1.04e+02    3.303676e-09  -1.699504e-05   9.999407e-01
   9 2.1367534228e+02  -1.53e+02  1.04e+01    3.303354e-09  -1.691756e-05   9.960589e-01
  10 1.9331484414e+02  -1.05e+04  1.04e+00    3.279991e-09  -1.130219e-05   7.148163e-01
  11 1.7083216620e+02  -1.32e+04  1.04e-01    3.220119e-09   3.087999e-06  -5.907913e-03
  12 1.7081739182e+02  -8.65e+00  1.04e-02    3.218545e-09   3.466367e-06  -2.485827e-02
  13 1.7081739182e+02  -5.98e-07  1.04e-03    3.218545e-09   3.466466e-06  -2.486325e-02
iter      chisq       delta/lim  lambda   a0             a1             a2

After 13 iterations the fit converged.
final sum of squares of residuals : 170.817
rel. change during last iteration : -5.98409e-12

degrees of freedom    (FIT_NDF)                        : 394
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 0.658443
variance of residuals (reduced chisquare) = WSSR/ndf   : 0.433547

Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a0              = 3.21854e-09      +/- 1.125e-11    (0.3496%)
a1              = 3.46647e-06      +/- 2.345e-06    (67.65%)
a2              = -0.0248633       +/- 0.1027       (413%)

correlation matrix of the fit parameters:
                a0     a1     a2
a0              1.000
a1             -0.969  1.000
a2              0.758 -0.874  1.000
```

en esta imagen mostramos la ejecución del ajuste realizado con los algoritmos de eficiencia O(n^2) y como vemos un buen ajuste porque el error cuando nos quedamos solo con el coeficiente de x^2 es muy cercano a 0, por lo tanto se confirma que el algoritmo burbuja pertenece a la familia cuadrática.

Si realizamos el mismo ajuste pero con una función cúbica obtenemos estos resultados:

*F(x) = A0*x^3+A1*x^2+A2*x+A3*

```
After 9 iterations the fit converged.
final sum of squares of residuals : 167.963
rel. change during last iteration : -6.83205e-06

degrees of freedom    (FIT_NDF)                        : 393
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 0.653748
variance of residuals (reduced chisquare) = WSSR/ndf   : 0.427386

Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a0              = 5.73726e-16      +/- 2.22e-16     (38.69%)
a1              = 3.04471e-09      +/- 6.818e-11    (2.239%)
a2              = 1.76333e-05      +/- 5.955e-06    (33.77%)
a3              = -0.273496        +/- 0.1402       (51.25%)

correlation matrix of the fit parameters:
                a0     a1     a2     a3
a0              1.000
a1             -0.986  1.000
a2              0.920 -0.970  1.000
a3             -0.686  0.767 -0.880  1.000
```

En esa imagen si nos fijamos en el coeficiente de la x^3 si ese es el único que vale 1 y los demás valen 0, tenemos un 38% de error, lo cual indica que se ajusta peor a esta familia que a la familia cuadrática

Por último hemos realizado la comprobación con los algoritmos de eficiencia *O(nlog(n))* ajustando a la función:

*F(x) = a*x*(log10(b*x)/log10(2))+c*

Y los valores obtenidos son los siguientes:

```
iter      chisq       delta/lim  lambda   a             b             c
   0 2.2639325488e+03   0.00e+00  4.46e+01    3.449998e-04   1.545649e-05   1.204515e+01
   1 2.2639325488e+03   0.00e+00  4.46e+00    3.449998e-04   1.545649e-05   1.204515e+01
iter      chisq       delta/lim  lambda   a             b             c

After 1 iterations the fit converged.
final sum of squares of residuals : 2263.93
rel. change during last iteration : 0

degrees of freedom    (FIT_NDF)                        : 394
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 2.39709
variance of residuals (reduced chisquare) = WSSR/ndf   : 5.74602

Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a                  = 0.000345      +/- 4.527e-06    (1.312%)
b                  = 1.54565e-05   +/- 2.743e-07    (1.775%)
c                  = 12.0451       +/- 0.5122       (4.253%)

correlation matrix of the fit parameters:
                a      b      c
a               1.000
b              -0.971  1.000
c               0.879 -0.953  1.000
```

como vemos el error es inmenso en comparación con los n^2 y los n^3, lo cual demuestra que el algoritmo burbuja no es tampoco nlog(n).

Seleccion:

En este caso hemos calculado la eficiencia híbrida del algoritmo seleccion, hemos hecho un ajuste a una función cuadrática y hemos calculado los coeficientes de la recta de regresión que más se adapta entre:

*F(x) = A0*x^2+A1*x+A2*

y la función que aparece al hacer la gráfica con los tiempos medidos por nosotros en uno de los ordenadores.

```
iter      chisq       delta/lim  lambda   a0            a1            a2
   0 1.2880294102e+23   0.00e+00  1.04e+10    1.000000e+00   1.000000e+00   1.000000e+00
   1 9.0651474300e+16  -1.42e+11  1.04e+09    8.326905e-04   9.999938e-01   1.000000e+00
   2 3.3459827542e+11  -2.71e+10  1.04e+08   -6.233873e-06   9.999934e-01   1.000000e+00
   3 3.3457118271e+11  -8.10e+00  1.04e+07   -6.240724e-06   9.999625e-01   1.000000e+00
   4 3.3251057204e+11  -6.20e+02  1.04e+06   -6.221472e-06   9.968783e-01   9.999999e-01
   5 1.9394361422e+11  -7.14e+04  1.04e+05   -4.751136e-06   7.613311e-01   9.999952e-01
   6 1.9013671641e+08  -1.02e+08  1.04e+04   -1.473873e-07   2.381196e-02   9.999805e-01
   7 4.8197371166e+01  -3.94e+11  1.04e+03    1.371345e-09  -1.911744e-05   9.999797e-01
   8 2.8343480082e+01  -7.00e+04  1.04e+02    1.419426e-09  -2.681974e-05   9.999511e-01
   9 2.8167006930e+01  -6.27e+02  1.04e+01    1.419189e-09  -2.676279e-05   9.970974e-01
  10 1.7163192866e+01  -6.41e+04  1.04e+00    1.402014e-09  -2.263464e-05   7.903412e-01
  11 5.0124485115e+00  -2.42e+05  1.04e-01    1.357999e-09  -1.205566e-05   2.604990e-01
  12 5.0044637111e+00  -1.60e+02  1.04e-02    1.356842e-09  -1.177750e-05   2.465676e-01
  13 5.0044637106e+00  -1.10e-05  1.04e-03    1.356842e-09  -1.177743e-05   2.465640e-01
iter      chisq       delta/lim  lambda   a0            a1            a2

After 13 iterations the fit converged.
final sum of squares of residuals : 5.00446
rel. change during last iteration : -1.10321e-10

degrees of freedom    (FIT_NDF)                        : 394
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 0.112702
variance of residuals (reduced chisquare) = WSSR/ndf   : 0.0127017

Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a0              = 1.35684e-09      +/- 1.926e-12    (0.1419%)
a1              = -1.17774e-05     +/- 4.014e-07    (3.408%)
a2              = 0.246564         +/- 0.01758      (7.128%)

correlation matrix of the fit parameters:
                a0     a1     a2
a0              1.000
a1             -0.969  1.000
a2              0.758 -0.874  1.000
```

en esta imagen mostramos la ejecución del ajuste realizado con los algoritmos de eficiencia O(n^2) y como vemos un buen ajuste porque el error cuando nos quedamos solo con el coeficiente de x^2 es muy cercano a 0, por lo tanto se confirma que el algoritmo burbuja pertenece a la familia cuadrática.

Si realizamos el mismo ajuste pero con una función cúbica obtenemos estos resultados:

*F(x) = A0*x^3+A1*x^2+A2*x+A3*

```
After 18 iterations the fit converged.
final sum of squares of residuals : 2.05134
rel. change during last iteration : -9.49837e-08

degrees of freedom    (FIT_NDF)                        : 393
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 0.0722474
variance of residuals (reduced chisquare) = WSSR/ndf   : 0.00521969

Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a0              = 5.83529e-16      +/- 2.453e-17    (4.204%)
a1              = 1.18003e-09      +/- 7.535e-12    (0.6386%)
a2              = 2.63151e-06      +/- 6.582e-07    (25.01%)
a3              = -0.00631751      +/- 0.01549      (245.2%)

correlation matrix of the fit parameters:
                a0     a1     a2     a3
a0              1.000
a1             -0.986  1.000
a2              0.920 -0.970  1.000
a3             -0.686  0.767 -0.880  1.000
```

En esa imagen si nos fijamos en el coeficiente de la x^3 si ese es el único que vale 1 y los demás valen 0, tenemos un 4,2% de error y el error sigue aumentando con las de mas variables, lo cual indica que se ajusta peor a esta familia que a la familia cuadrática

Inserción

Para este algoritmo la función que se ha usado a ajustar es la siguiente

F(x) = A0*x^2+A1*x+A2

y hemos obtenido los siguientes resultados:

```
iter      chisq       delta/lim  lambda   a0             a1             a2
   0 1.2880294029e+23  0.00e+00 1.05e+10  1.000000e+00  1.000000e+00  1.000000e+00
   1 9.3452546011e+16 -1.38e+11 1.05e+09  8.455523e-04  9.999938e-01  1.000000e+00
   2 3.3452923522e+11 -2.79e+10 1.05e+08 -6.234524e-06  9.999935e-01  1.000000e+00
   3 3.3450206233e+11 -8.12e+00 1.05e+07 -6.241596e-06  9.999630e-01  1.000000e+00
   4 3.3247326236e+11 -6.10e+02 1.05e+06 -6.222637e-06  9.969259e-01  9.999999e-01
   5 1.9533191827e+11 -7.02e+04 1.05e+05 -4.769491e-06  7.641322e-01  9.999953e-01
   6 1.9730432254e+08 -9.89e+07 1.05e+04 -1.510955e-07  2.426659e-02  9.999806e-01
   7 5.9735884121e+01 -3.30e+11 1.05e+03  4.548452e-10 -1.173796e-05  9.999798e-01
   8 3.8487927029e+01 -5.52e+04 1.05e+02  5.045892e-10 -1.970671e-05  9.999468e-01
   9 3.8250780848e+01 -6.20e+02 1.05e+01  5.043177e-10 -1.964135e-05  9.966634e-01
  10 2.2600187673e+01 -6.92e+04 1.05e+00  4.838387e-10 -1.471262e-05  7.491517e-01
  11 2.0201290637e+00 -1.02e+06 1.05e-01  4.229933e-10 -6.884114e-08  1.376931e-02
  12 2.0019460749e+00 -9.08e+02 1.05e-02  4.211308e-10  3.794257e-07 -8.741795e-03
  13 2.0019460732e+00 -8.51e-05 1.05e-03  4.211302e-10  3.795630e-07 -8.748688e-03
iter      chisq       delta/lim  lambda   a0             a1             a2

After 13 iterations the fit converged.
final sum of squares of residuals : 2.00195
rel. change during last iteration : -8.51101e-010

degrees of freedom    (FIT_NDF)                        : 388
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 0.0718307
variance of residuals (reduced chisquare) = WSSR/ndf   : 0.00515965

Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a0              = 4.2113e-010      +/- 1.275e-012   (0.3028%)
a1              = 3.79563e-007     +/- 2.692e-007   (70.93%)
a2              = -0.00874869      +/- 0.012        (137.1%)

correlation matrix of the fit parameters:
                a0     a1     a2
a0              1.000
a1             -0.971  1.000
a2              0.778 -0.887  1.000
```

Como podemos observar vemos que el error de x^2 es muy cercano a 0, esto quiere decir que el ajuste realizado es bastante bueno.

Para un ajuste lineal la función a ajustar es F(x) = A0*x+A1

y se obtienen los siguientes resultados:

```
iter      chisq       delta/lim  lambda   a0             a1
   0 2.3062834170e+04  0.00e+00 3.48e-05  4.211302e-10  3.795630e-07
   1 1.4160987096e+03 -1.53e+06 3.48e-06  6.372608e-05 -3.263105e-02
   2 7.6166088710e+02 -8.59e+04 3.48e-07  7.569571e-05 -1.620587e+00
   3 5.6476123417e+02 -3.49e+04 3.48e-08  8.661012e-05 -3.078440e+00
   4 5.6474463709e+02 -2.94e+00 3.48e-09  8.671124e-05 -3.091947e+00
   5 5.6474463709e+02 -2.52e-08 3.48e-10  8.671125e-05 -3.091948e+00
iter      chisq       delta/lim  lambda   a0             a1

After 5 iterations the fit converged.
final sum of squares of residuals : 564.745
rel. change during last iteration : -2.52439e-013

degrees of freedom    (FIT_NDF)                        : 389
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 1.2049
variance of residuals (reduced chisquare) = WSSR/ndf   : 1.45179

Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a0              = 8.67113e-005     +/- 1.08e-006    (1.245%)
a1              = -3.09195         +/- 0.1263       (4.086%)

correlation matrix of the fit parameters:
                a0     a1
a0              1.000
a1             -0.876  1.000
```

En este caso el error de x es cercano a 1, por tanto el ajuste no es muy bueno.

Para un ajuste n^3 la función a ajustar es F(x) = A0*x^3+A1*x^2+A2*x+A3

y se obtienen los siguientes resultados:

```
iter      chisq       delta/lim  lambda   a0             a1             a2             a3
   0 1.7444848821e+25  0.00e+00 1.36e+11  8.671125e-05 -3.091948e+00 -8.748688e-03  1.000000e+00
   1 5.9837513887e+21 -2.91e+08 1.36e+10  7.580905e-06 -1.292134e+00 -8.748688e-03  1.000000e+00
   2 1.8861768204e+17 -3.17e+09 1.36e+09  4.235093e-08 -7.260519e-03 -8.748688e-03  1.000000e+00
   3 6.0599684023e+08 -3.11e+13 1.36e+08  1.935185e-12 -2.788619e-07 -8.748688e-03  1.000000e+00
   4 4.0577595404e+06 -1.48e+07 1.36e+07 -4.573024e-13  1.313048e-07 -8.748688e-03  1.000000e+00
   5 4.0577593609e+06 -4.42e-03 1.36e+06 -4.573037e-13  1.313051e-07 -8.748688e-03  1.000000e+00
iter      chisq       delta/lim  lambda   a0             a1             a2             a3

After 5 iterations the fit converged.
final sum of squares of residuals : 4.05776e+006
rel. change during last iteration : -4.42357e-008

degrees of freedom    (FIT_NDF)                       : 387
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 102.397
variance of residuals (reduced chisquare) = WSSR/ndf   : 10485.2

Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a0              = -4.57304e-013    +/- 3.667e-014   (8.02%)
a1              = 1.31305e-007     +/- 1.142e-008   (8.699%)
a2              = -0.00874869      +/- 0.001021     (11.67%)
a3              = 1                +/- 24.79        (2479%)

correlation matrix of the fit parameters:
                a0     a1     a2     a3
a0              1.000
a1             -0.987  1.000
a2              0.927 -0.973  1.000
a3             -0.724  0.800 -0.901  1.000
```

Como vemos el error x^3 es de un 8.02% y deducimos que el ajuste es muy malo.

Mergesort:

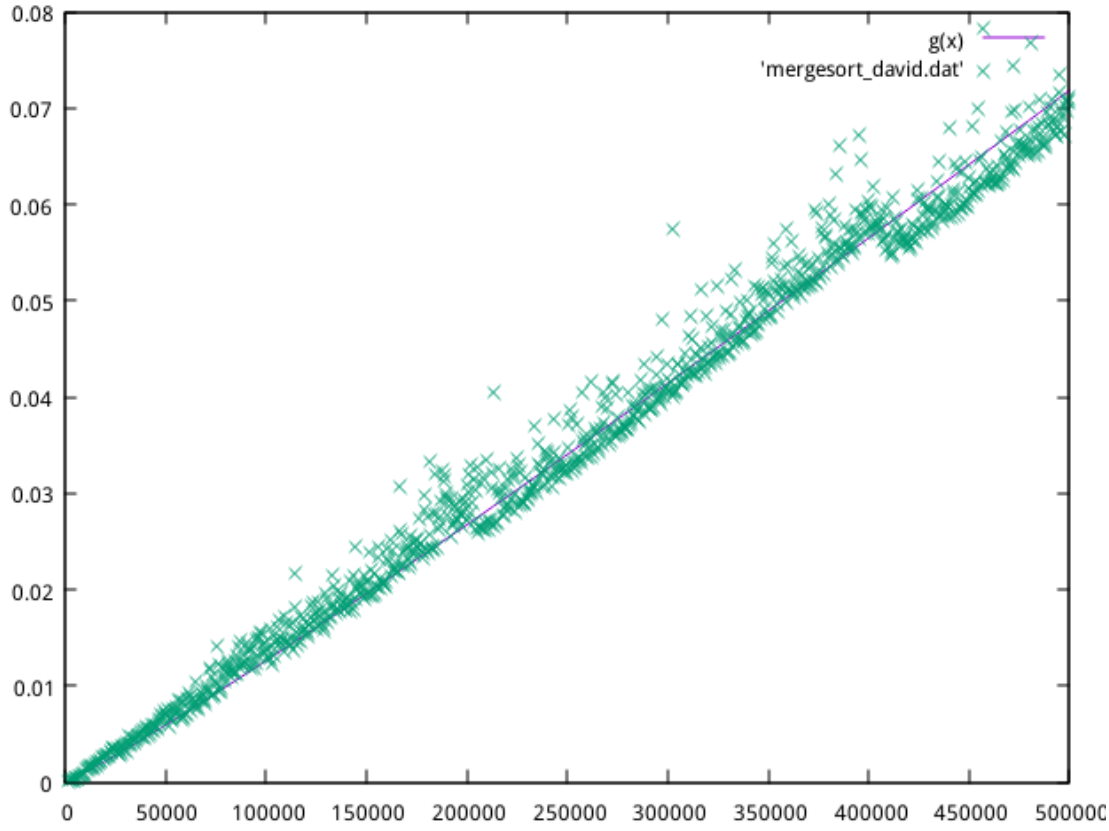
Para este algoritmo la función que se ha usado a ajustar es la siguiente

*F(x) = a*x*(log10(b*x)/log10(2))*

y hemos obtenido los siguientes resultados.

```
Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a                 = 7.63663e-09    +/- 5.514e-10    (7.221%)
b                 = 0.92304        +/- 0.8478       (91.85%)

f(x)  = a*x*(log10(b*x)/log10(2))
```

A continuación se muestra una gráfica comparando los tiempos medidos con la función ajustada.



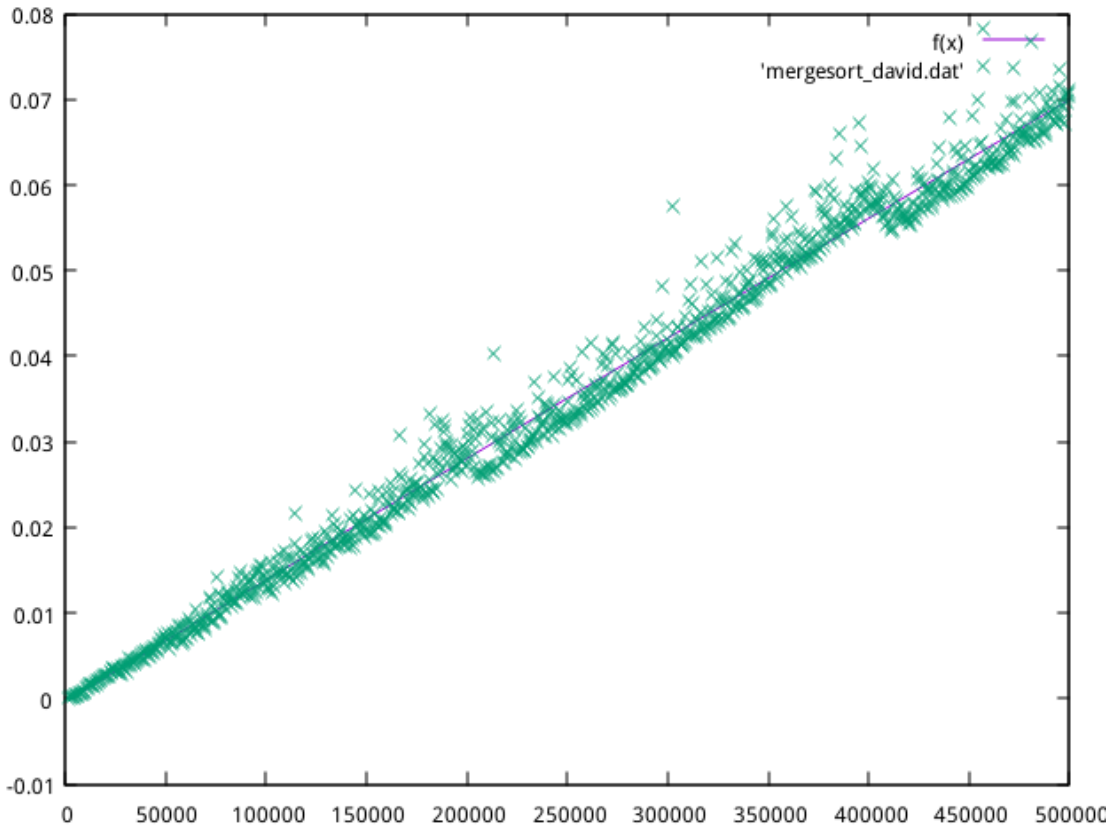
Se puede apreciar que se ajusta perfectamente ya que el algoritmo pertenece a esa familia y para otras familias se ha obtenido lo siguiente.

Para los tiempos anteriores se ha ajustado a la función cuadrática *F(x) = a*x^2 + b*x + c* y

obtenemos lo siguiente.

```
Final set of parameters            Asymptotic Standard Error
=======================            =========================
a                 = -4.90714e-15   +/- 3.774e-15    (76.9%)
b                 = 1.43255e-07    +/- 1.956e-09    (1.365%)
c                 = -0.000433595   +/- 0.0002126    (49.04%)

f(x) = a*x*x+b*x+c
```

Al comparar en la gráfica esta función ajustada a los tiempos que teníamos se observa lo siguiente.



Los función se ajusta a la nube de puntos a pesar de que no sea su familia aunque podemos ver que tiene un alto índice de error por lo que se demuestra que esa función no es la óptima con la que representar los datos de la nube de puntos.

Quicksort:

Para este algoritmo la función que se ha usado a ajustar es la siguiente

*F(x) = f(x)=a*x*(log10(x)/log10(2))+b*x*

degrees of freedom (FIT_NDF) : 995

rms of residuals (FIT_STDFIT) = sqrt(WSSR/ndf) : 0.00202018

variance of residuals (reduced chisquare) = WSSR/ndf : 4.08113e-06

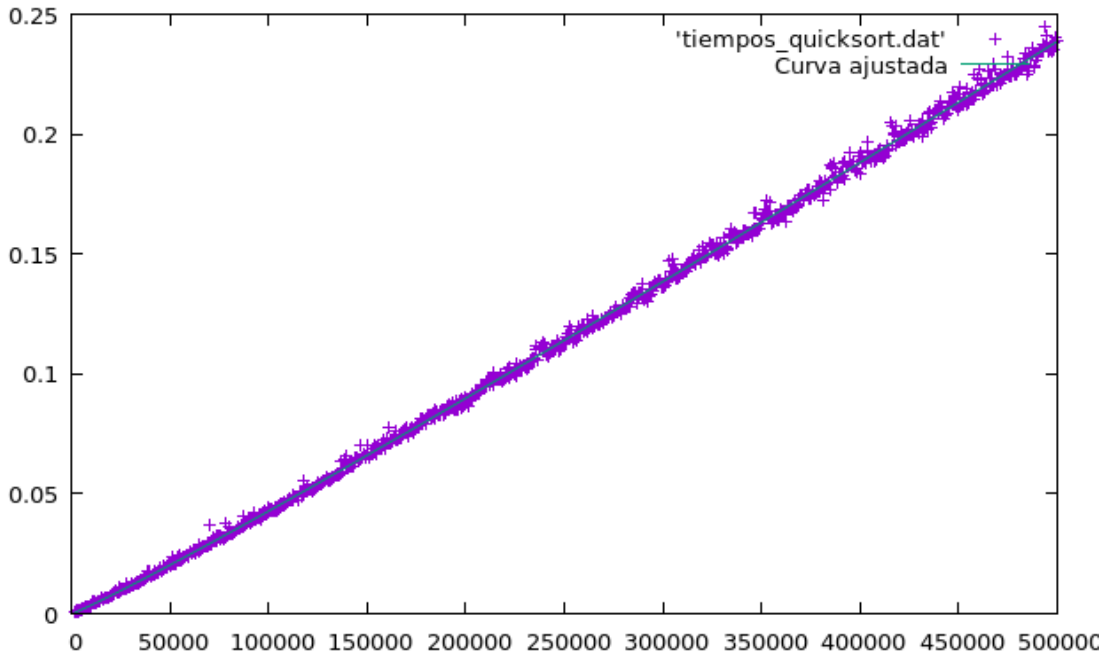
Final set of parameters		Asymptotic Standard Error
a	= 2.11356e-08	+/- 4.598e-10 (2.176%)
b	= 7.72479e-08	+/- 8.488e-09 (10.99%)

correlation matrix of the fit parameters:

	a	b
a	1.000	
b	-1.000	1.000

Como se puede ver el error es muy bajo y muestra lo exacto que es su designación en la familia de los nlogn

A continuación presentamos la gráfica de los tiempos tomados con el ajuste



A continuación las regresiones para O(n) y O(n^3)

degrees of freedom (FIT_NDF) : 993

rms of residuals (FIT_STDFIT) = sqrt(WSSR/ndf) : 0.00202367

variance of residuals (reduced chisquare) = WSSR/ndf : 4.09524e-06

Final set of parameters		Asymptotic Standard Error
a	= -1.43476e-19	+/- 2.738e-20 (19.08%)
b	= 1.86762e-13	+/- 2.09e-14 (11.19%)
c	= 4.21195e-07	+/- 4.524e-09 (1.074%)
d	= -0.00083145	+/- 0.0002632 (31.65%)

correlation matrix of the fit parameters:

	a	b	c	d
a	1.000			
b	-0.986	1.000		
c	0.918	-0.969	1.000	
d	-0.672	0.754	-0.872	1.000

Como podemos ver el porcentaje de error en la primera y ultima constante es bastante alto y eso nos muestra que no es factible como O(n^3).

degrees of freedom (FIT_NDF) : 995

rms of residuals (FIT_STDFIT) = sqrt(WSSR/ndf) : 0.00251606

variance of residuals (reduced chisquare) = WSSR/ndf : 6.33057e-06

Final set of parameters		Asymptotic Standard Error
a	= 4.82484e-07	+/- 5.537e-10 (0.1148%)
b	= -0.00508738	+/- 0.0001602 (3.149%)

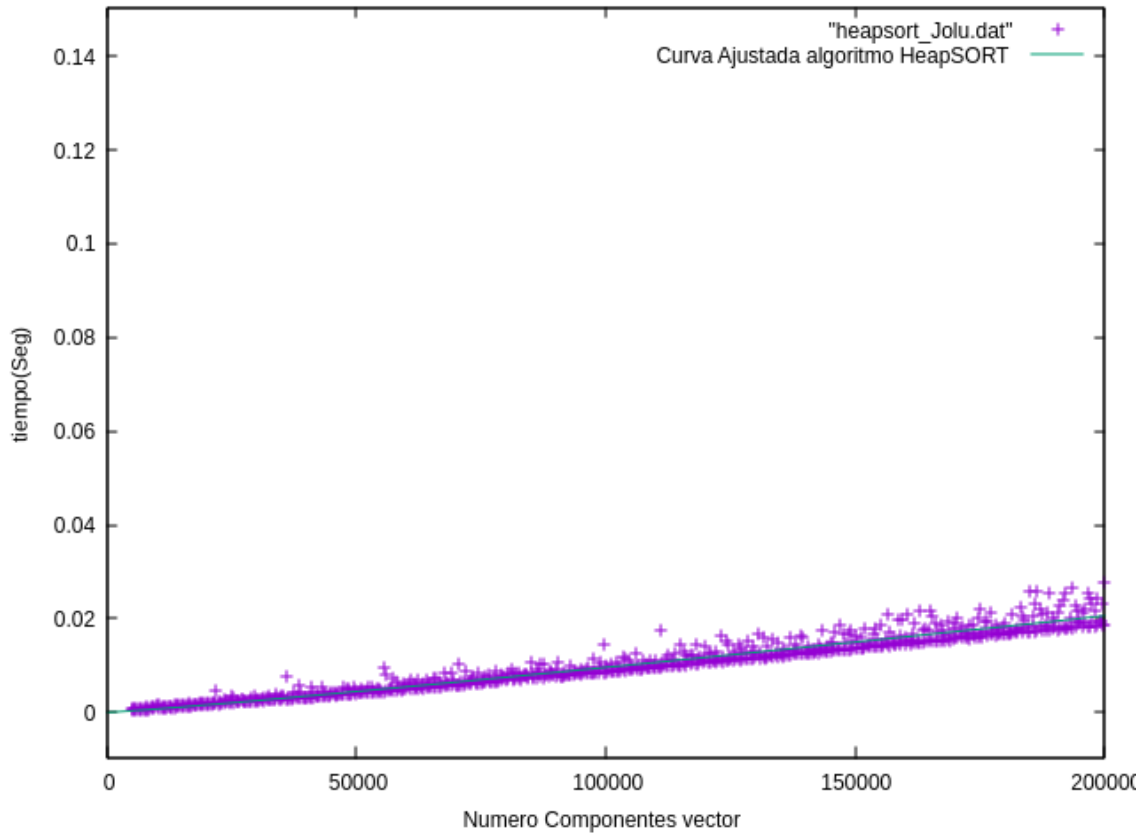
correlation matrix of the fit parameters:

	a	b

a	1.000
b	-0.868 1.000

Como podemos ver en el porcentaje de error tan bajo el quicksort es un algoritmo muy cercano a O(n).

Heapsort:



En esta gráfica se puede ver el ajuste del algoritmo de Heapsort según la función a*x*log(b*x)+c*x.

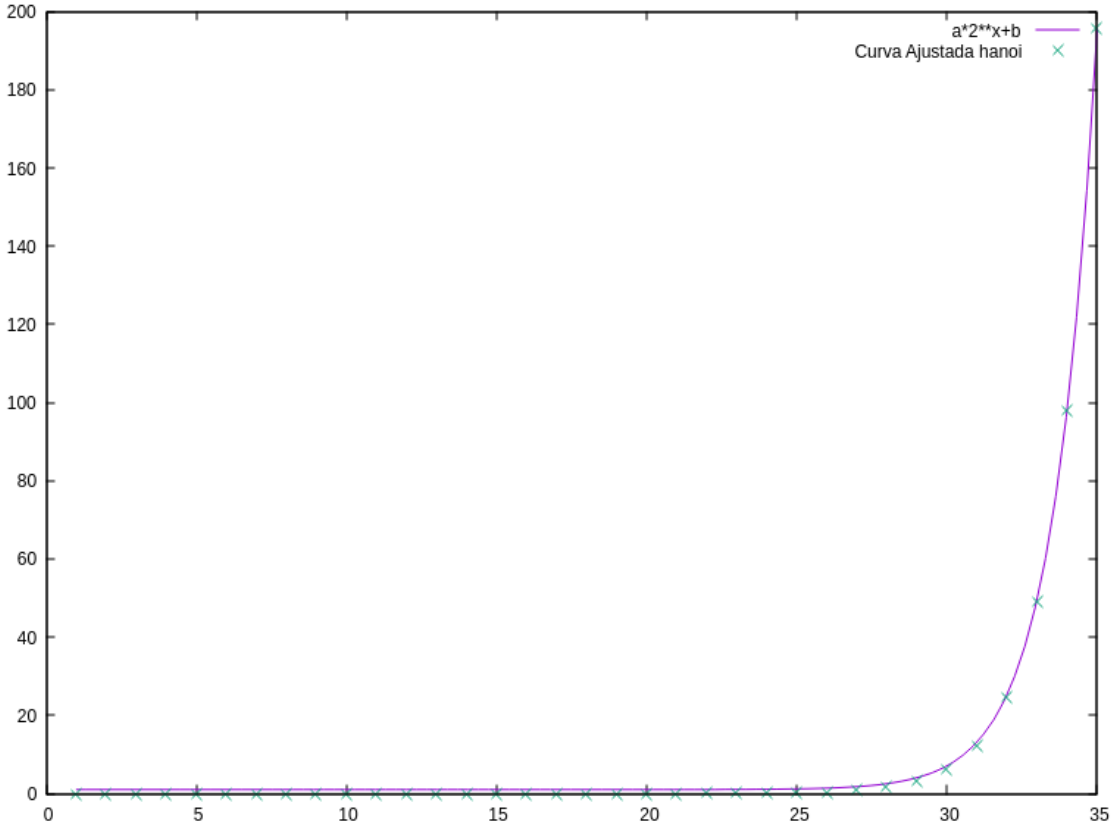
El ajuste es adecuado porque como podemos apreciar, excepto en en coeficiente a que obtenemos un porcentaje de error de 13.03%, en los demás coeficientes apenas llegamos al 0%.

```
gnuplot> fit a*x*(log10(b*x)/log(2))+c*x 'heapsort_Jolu.dat' via a,b,c
iter      chisq       delta/lim  lambda   a             b             c
   0 1.5979006538e-03   0.00e+00  7.67e-03    1.609976e-08   4.773788e-01  -1.267180e-08
   1 1.5979006538e-03   0.00e+00  7.67e-04    1.609976e-08   4.773788e-01  -1.267180e-08
iter      chisq       delta/lim  lambda   a             b             c

After 1 iterations the fit converged.
final sum of squares of residuals : 0.0015979
rel. change during last iteration : 0

degrees of freedom    (FIT_NDF)                        : 779
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 0.00143221
variance of residuals (reduced chisquare) = WSSR/ndf   : 2.05122e-06

Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a               = 1.60998e-08      +/- 2.098e-09    (13.03%)
b               = 0.477379         +/- 1.351e+10    (2.831e+12%)
c               = -1.26718e-08     +/- 285.4        (2.252e+12%)

correlation matrix of the fit parameters:
                a      b      c
a               1.000
b              -0.012  1.000
c               0.012 -1.000  1.000
```

HANOI

los valores que hemos tomado para representar hanoi han sido de 1 en 1 hasta 35, al ser del

orden 2^n, es decir exponencial, se va muy rápido a tiempos de ejecución infinitos, el máximo de valores que le hemos dado ha sido 35.

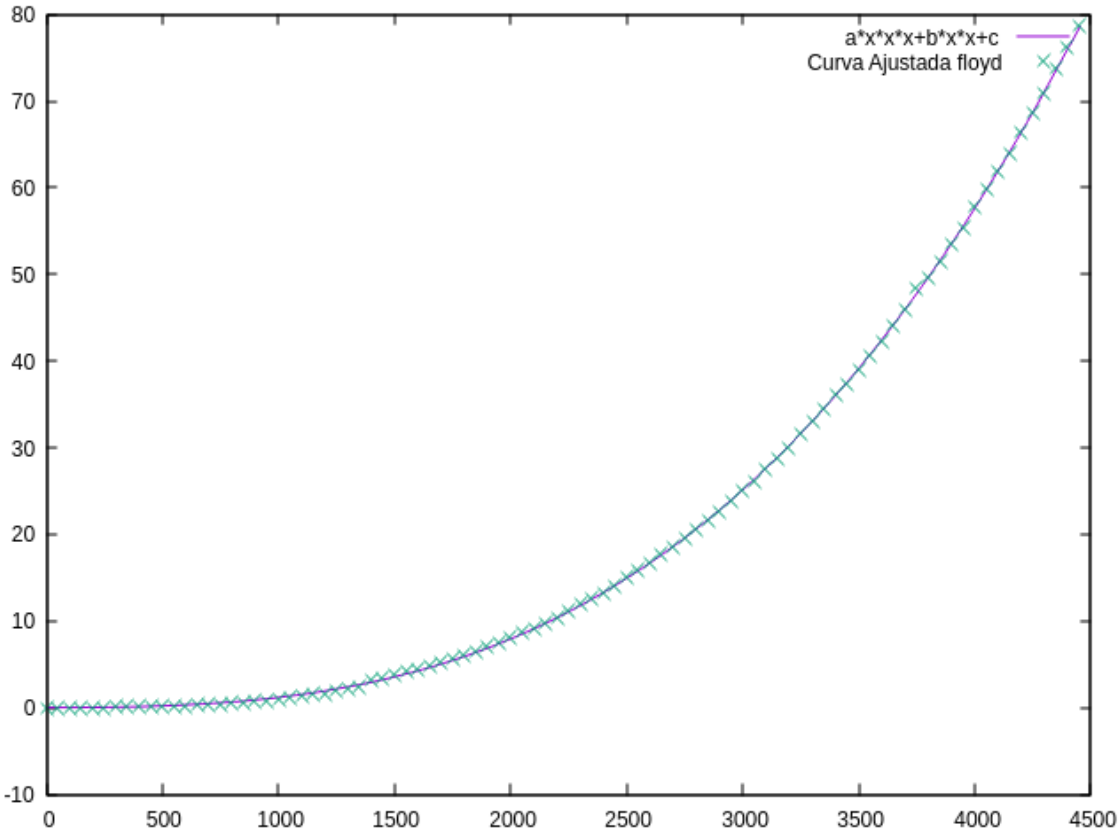


En la siguiente imagen hemos hecho el ajuste al algoritmo de hanoi con una función 2^n y como podemos ver el ajuste es bueno, dado que el error del ajuste al coeficiente a es muy bajito.

```
gnuplot> fit a*2**x+b 'hanoi_Jolu.dat' via a,b
iter      chisq       delta/lim  lambda   a             b
   0 1.5741221431e+21   0.00e+00  4.74e+09    1.000000e+00   1.000000e+00
   1 3.1226386493e+17  -5.04e+08  4.74e+08    1.408451e-02   1.000000e+00
   2 6.3709115670e+09  -4.90e+12  4.74e+07    2.017443e-06   1.000000e+00
   3 3.1447514098e+01  -2.03e+13  4.74e+06    5.660955e-09   1.000000e+00
   4 3.1434512277e+01  -4.14e+01  4.74e+05    5.658081e-09   1.000000e+00
   5 3.1434512268e+01  -2.84e-05  4.74e+04    5.658081e-09   1.000000e+00
iter      chisq       delta/lim  lambda   a             b

After 5 iterations the fit converged.
final sum of squares of residuals : 31.4345
rel. change during last iteration : -2.84428e-10

degrees of freedom    (FIT_NDF)                        : 33
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 0.975992
variance of residuals (reduced chisquare) = WSSR/ndf   : 0.952561

Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a               = 5.65808e-09      +/- 2.573e-11    (0.4547%)
b               = 1                +/- 0.1725       (17.25%)

correlation matrix of the fit parameters:
                a      b
a               1.000
b              -0.293  1.000
```

FLOYD

En esta imagen representamos los valores que hemos tomado para ejecutar el algoritmo de floyd, como es un algoritmo perteneciente a la familia de los cúbicos, no hemos podido dar una gran cantidad de valores para representarlo en la gráfica dado a que tardaba demasiado.



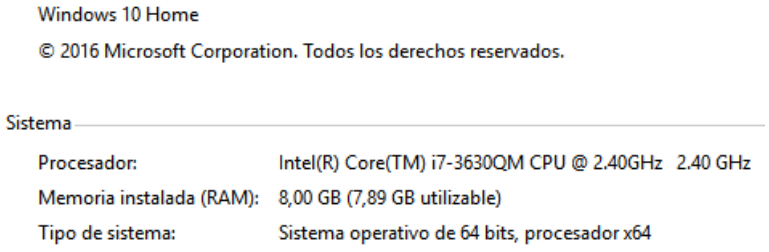

```
gnuplot> fit a*x*x*x+b*x*x+c 'floyd_Jolu.dat' via a,b,c
iter      chisq       delta/lim  lambda   a             b             c

   0 2.4280363318e+00   0.00e+00  1.59e+01    8.101710e-10   3.601051e-07  -2.3
06394e-05
   1 2.4280363318e+00  -2.05e-07  1.59e+00    8.101710e-10   3.601051e-07  -2.3
06394e-05
iter      chisq       delta/lim  lambda   a             b             c


After 1 iterations the fit converged.
final sum of squares of residuals : 2.42804
rel. change during last iteration : -2.04501e-12

degrees of freedom    (FIT_NDF)                       : 87
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 0.167058
variance of residuals (reduced chisquare) = WSSR/ndf   : 0.0279085

Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a               = 8.10171e-10      +/- 4.15e-12     (0.5122%)
b               = 3.60105e-07      +/- 1.764e-08    (4.898%)
c               = -2.30639e-05     +/- 0.03494      (1.515e+05%)

correlation matrix of the fit parameters:
                a      b      c
a               1.000
b              -0.986  1.000
c               0.658 -0.742  1.000
```

Podemos ver que el error del coeficiente a es próximo a 0 y podemos decir que el ajuste realizado es bueno.

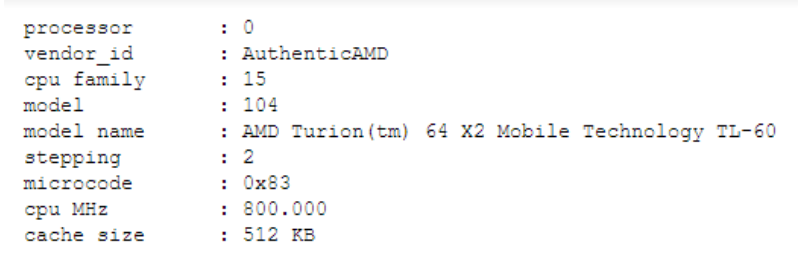
4. Otro aspecto interesante a analizar mediante este tipo de estudio es la variación de la eficiencia empírica en función de parámetros externos tales como: las opciones de compilación utilizada (con/sin optimización), el ordenador donde se realizan las pruebas, el sistema operativo, etc. Sugiera algún estudio de este tipo, consulte con el profesor de prácticas y llévelo a cabo.

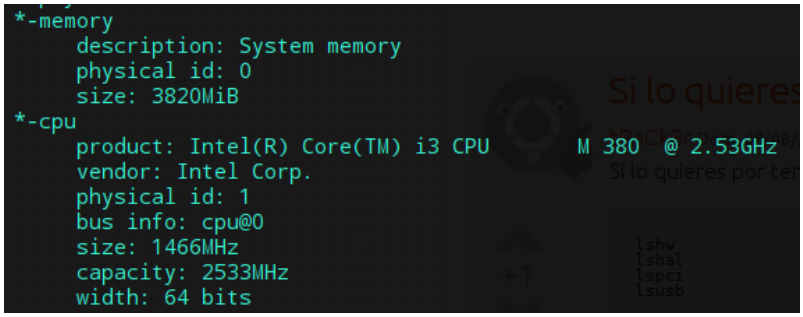
Burbuja: (explicado en el punto nº1 con gráfica y comparativas)

El hardware y los sistemas operativos que hemos usado para realizar esta práctica son:



Linux:





```
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 60
model name      : Intel(R) Core(TM) i5-4210H CPU @ 2.90GHz
stepping        : 3
microcode       : 0x1c
cpu MHz         : 2899.433
cache size      : 3072 KB
physical id     : 0
siblings        : 4
core id         : 0
cpu cores       : 2
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
wp              : yes
```