



ALGORÍTMICA

Práctica1. Eficiencia



Grado Ingeniería Informática

Prácticas Algorítmica A2

Heredia Castillo, Antonio Jesús

Hinojosa Castro, Miguel Ángel

Luque Callejas, David

Pérez Lendínez, José Manuel

Rodriguez Vílchez, Pedro Javier

Índice

1. Cálculo de la eficiencia empírica	2
1.1. Archivos utilizados.....	2
1.2. Tamaños utilizados para cada algoritmo	2
1.3. Resultados empíricos de los algoritmos de orden $O(n^2)$	3
1.4. Resultados empíricos de los algoritmos de $O(n \log n)$	3
1.5. Resultados empíricos obtenidos con el algoritmo de $O(n^3)$	4
1.6. Resultados empíricos obtenidos con el algoritmo de $O(2n)$	4
2. Cálculo de la eficiencia híbrida	5
3. Gráficas obtenidas.....	6
3.1. Selección.....	6
3.2. Burbuja	6
3.3. Inserción	7
3.4. Conjunto de algoritmos n^2	7
3.5. Heapsort.....	8
3.6. Mergesort.....	8
3.7. Quicksort	9
3.8. Conjunto de algoritmos n^3	9
3.9. Hanoi	10
3.10. Floyd	10
4. Ajustes que no corresponden con la eficiencia teórica	11
4.1. Inserción (orden cúbico).....	11
4.2. Burbuja (lineal)	12
4.3. Selección ($n \log n$)	12
4.4. Hanoi (cuadrático).....	13
4.5. Floyd (cúbico)	13
4.6. Heapsort ($\log n$).....	14
4.7. Mergesort($\log n$)	14
4.8. Quicksort ($\log n$).....	15
5. Datos empíricos en otro ordenador.....	15

1. Cálculo de la eficiencia empírica

1.1. Archivos utilizados

Para obtener los datos empíricos de cada algoritmo hemos añadido a cada uno las funciones correspondientes para calcular el tiempo de ejecución y hemos creado un script que contiene un bucle y en cada ejecución del mismo ejecuta el problema con un tamaño cada vez mayor.

```
clock_t tini;    // Anotamos el tiempo de inicio
clock_t tfin;    // Anotamos el tiempo de finalización

tini=clock();

burbuja(T, n);

tfin=clock();
cout << n << "\t" << (double)( tfin-tini ) / CLOCKS_PER_SEC << endl;
```

Ejemplo función para obtener tiempos de ejecución

```
#!/bin/bash
# Variables:
INICIO=1000
FIN=30000
INCREMENTO=1000
EJECUTABLE=burbuja
SALIDA=burbuja.data

i=$INICIO
echo > $SALIDA
while [ $i -le $FIN ]
do
    echo Ejecución tam = $i
    echo `./$EJECUTABLE $i` >> $SALIDA
    i=$((i+$INCREMENTO))
done
```

Ejemplo Script.sh

1.2. Tamaños utilizados para cada algoritmo

En los algoritmos de orden n^2 hemos utilizado 30 valores empezando en 1000.

Con los de eficiencia $n \log n$ se aumenta de 20000 en 20000.

En el caso de los demás algoritmos hemos usado valores más pequeños, para Floyd usamos valores de 40 en 40 empezando en 1 y en Hanoi de 10 a 35.

1.3. Resultados empíricos de los algoritmos de orden $O(n^2)$

Tabla con los tiempos de ejecución obtenidos en los diferentes algoritmos de orden n cuadrado.

N	Burbuja	Insercción	Selección	N	Burbuja	Insercción	Selección
1000	0,00375	0,002477	0,001334	16000	0,597873	0,233063	0,279839
2000	0,007292	0,004216	0,005329	17000	0,677438	0,259643	0,314333
3000	0,015840	0,008890	0,010958	18000	0,770653	0,291495	0,351419
4000	0,029313	0,015784	0,018997	19000	0,900122	0,328350	0,392033
5000	0,048540	0,023701	0,029772	20000	0,956463	0,343986	0,440533
6000	0,073328	0,034037	0,043400	21000	1,067130	0,393242	0,490400
7000	0,101383	0,045539	0,057279	22000	1,183700	0,423758	0,524967
8000	0,139372	0,060141	0,074252	23000	1,294400	0,462558	0,572054
9000	0,180604	0,072382	0,092402	24000	1,432060	0,505952	0,623997
10000	0,218605	0,090540	0,111347	25000	1,540350	0,544577	0,675133
11000	0,276473	0,106726	0,140968	26000	1,648840	0,592582	0,728445
12000	0,335401	0,130813	0,163718	27000	1,775270	0,642469	0,806852
13000	0,401327	0,148093	0,191532	28000	1,916830	0,680235	0,845424
14000	0,450787	0,171633	0,214461	29000	2,058860	0,742371	0,927268
15000	0,523129	0,199231	0,257192	30000	2,265890	0,783000	0,970406

1.4. Resultados empíricos de los algoritmos de $O(n \log n)$

Tabla con los resultados empíricos que tienen orden $n \times \log n$.

N	Mergesort	Heapsort	Quicksort	N	Mergesort	Heapsort	Quicksort
100000	0,020998	0,020079	0,013042	2700000	0,601968	0,651631	0,398498
300000	0,059492	0,060126	0,039984	2900000	0,660896	0,718083	0,428
500000	0,100641	0,105166	0,07166	3100000	0,718868	0,786469	0,45299
700000	0,148348	0,149675	0,095798	3300000	0,678016	0,819325	0,486722
900000	0,178828	0,194343	0,126597	3500000	0,723496	0,886464	0,519231
1100000	0,232415	0,246041	0,156967	3700000	0,80633	0,953508	0,572566
1300000	0,2858	0,283763	0,194075	3900000	0,846657	1,01989	0,580353
1500000	0,338734	0,343653	0,214074	4100000	0,882552	1,08871	0,606713
1700000	0,337637	0,379137	0,243824	4300000	0,93975	1,14813	0,633054
1900000	0,391266	0,426653	0,272155	4500000	0,994608	1,20958	0,683602
2100000	0,457877	0,485304	0,308648	4700000	1,038	1,25358	0,701406
2300000	0,500484	0,540172	0,329448	4900000	1,09599	1,31249	0,751409
2500000	0,54425	0,589261	0,362825				

1.5. Resultados empíricos obtenidos con el algoritmo de $O(n^3)$

Recopilación de los datos empíricos obtenidos por el algoritmo Floyd de orden n cúbico.

N	Floyd	N	Floyd
1	3,00E-06	521	0,636119
41	0,002041	561	0,796567
81	0,003428	601	0,970594
121	0,009775	641	1,17744
161	0,021344	681	1,40977
201	0,03993	721	1,67106
241	0,068375	761	1,96548
281	0,10492	801	2,29842
321	0,159368	841	2,64581
361	0,225831	881	3,07501
401	0,293284	921	3,47293
441	0,389772	961	3,95055
481	0,504097		

1.6. Resultados empíricos obtenidos con el algoritmo de $O(2n)$

Datos empíricos obtenidos del algoritmo Hanoi.

N	Hanoi	N	Hanoi
10	2,90E-05	23	0,046509
11	1,50E-05	24	0,086428
12	2,40E-05	25	0,17018
13	4,60E-05	26	0,353614
14	9,70E-05	27	0,726802
15	0,000194	28	1,38496
16	0,000436	29	2,70943
17	0,000767	30	5,37542
18	0,001544	31	10,6737
19	0,00313	32	21,3106
20	0,006338	33	42,6922
21	0,012307	34	85,1978
22	0,024984	35	170,425

2. Cálculo de la eficiencia híbrida

Para calcular la eficiencia híbrida de los algoritmos hemos utilizado la función de Gnuplot que nos permite hacer la regresión lineal y obtener así el ajuste a la función. Para ello hemos generado varias gráficas, individuales, donde mostramos los datos de cada algoritmo por separado, y múltiples, que usamos para comparar varios algoritmos en una misma gráfica.

Para las gráficas individuales de cada algoritmo hemos utilizado Gnuplot siguiendo los siguientes pasos:

1. Escribimos la función que vamos a dibujar. Por ejemplo: con el algoritmo de la burbuja hemos utilizado el comando: $f(x) = a \cdot x \cdot x$
2. Ajustamos la función a los datos empíricos que hemos obtenidos anteriormente, obteniendo una constante y un porcentaje de error. Usamos el comando:

Fit $f(x)$ 'burbuja.data' via a

3. Dibujamos con el comando plot:

plot [datos_inicio:datos_fin] $f(x)$, 'burbuja.data'

Para las gráficas múltiples hemos seguido los siguientes pasos en Gnuplot:

1. Escribimos una función por cada algoritmo que vamos a dibujar. Ej:
 $f(x) = a \cdot x \cdot x$
 $g(x) = b \cdot x \cdot x$
2. Ajustamos las funciones anteriores a los datos empíricos que hemos obtenido. Ej:

Fit $f(x)$ 'burbuja.data' via a

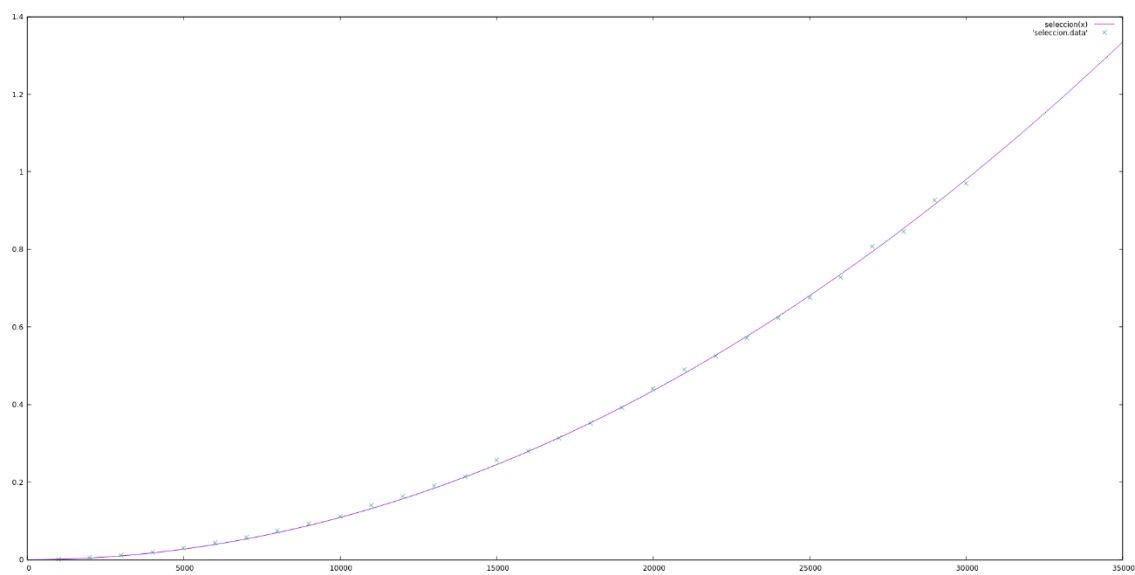
Fit $g(x)$ 'inserccion.data' via b

3. Dibujamos con el comando plot:

plot [datos_inicio:datos_fin] $f(x), g(x)$

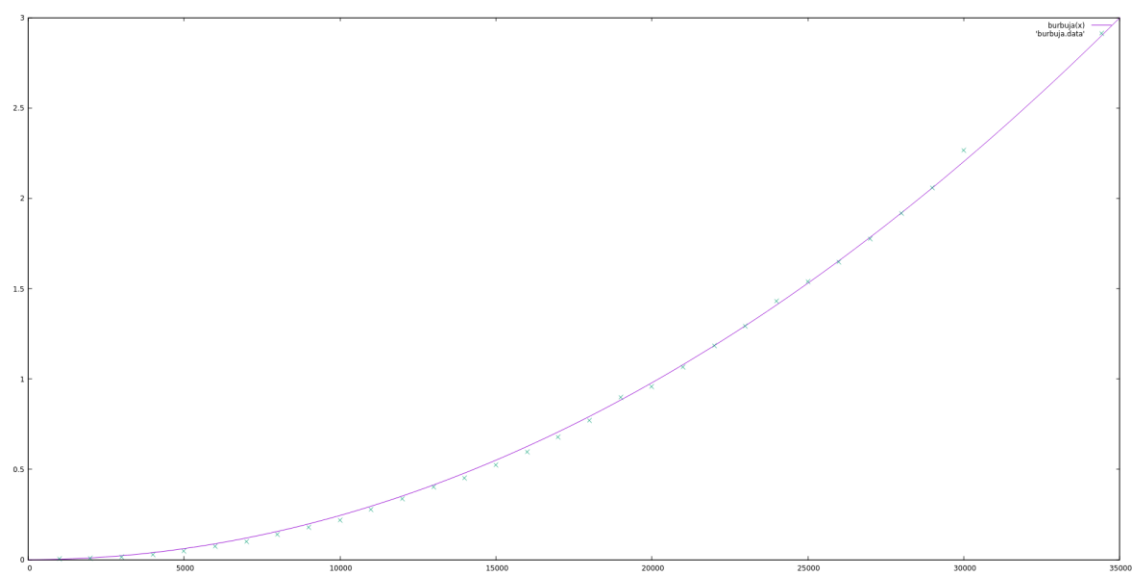
3. Gráficas obtenidas

3.1. Selección



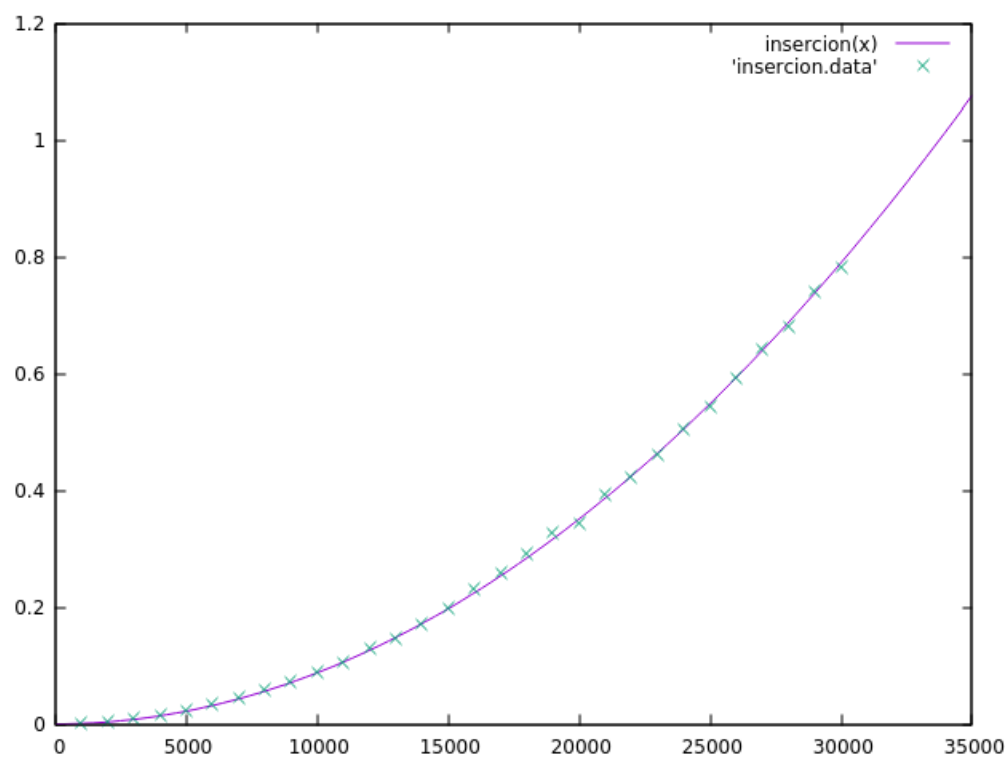
Constante	% error
1.08966e-09	0.2504%

3.2. Burbuja



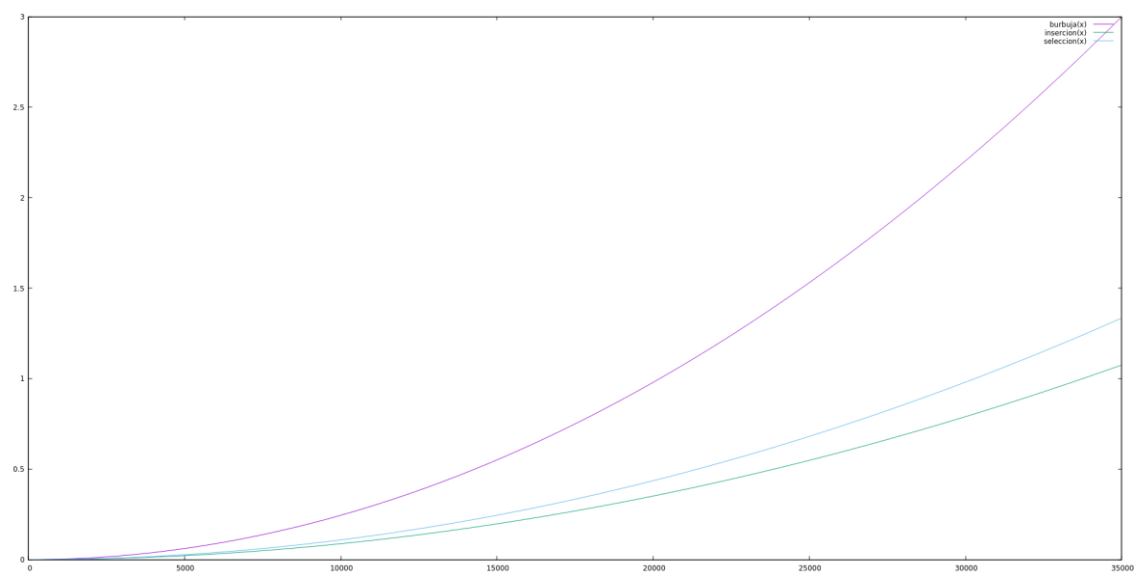
Constante	%error
2.44877e-09	0.3683%

3.3. Inserción



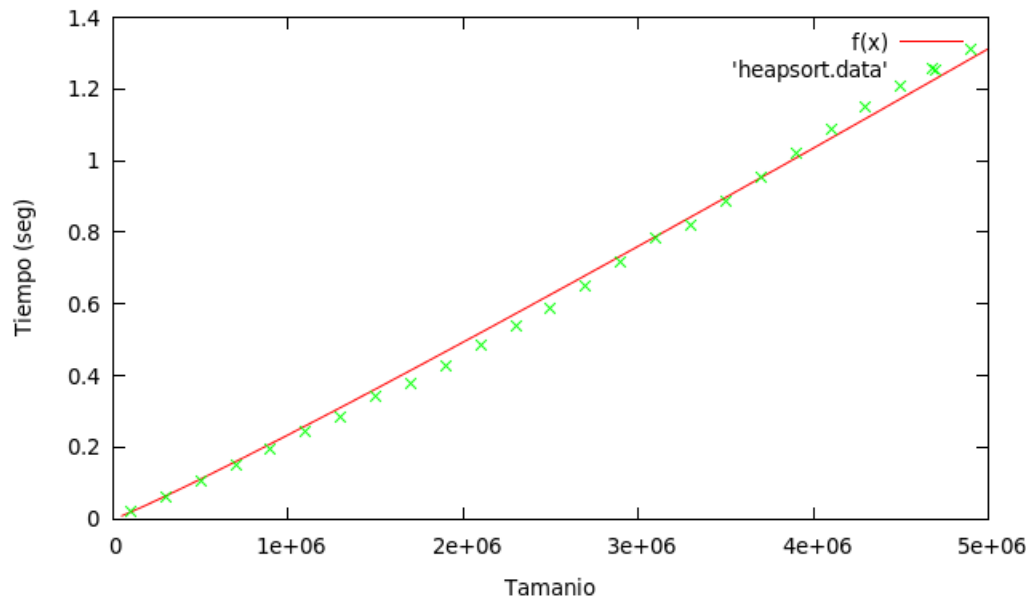
Constante	%error
8.77886e-10	0.2263%

3.4. Conjunto de algoritmos n^2



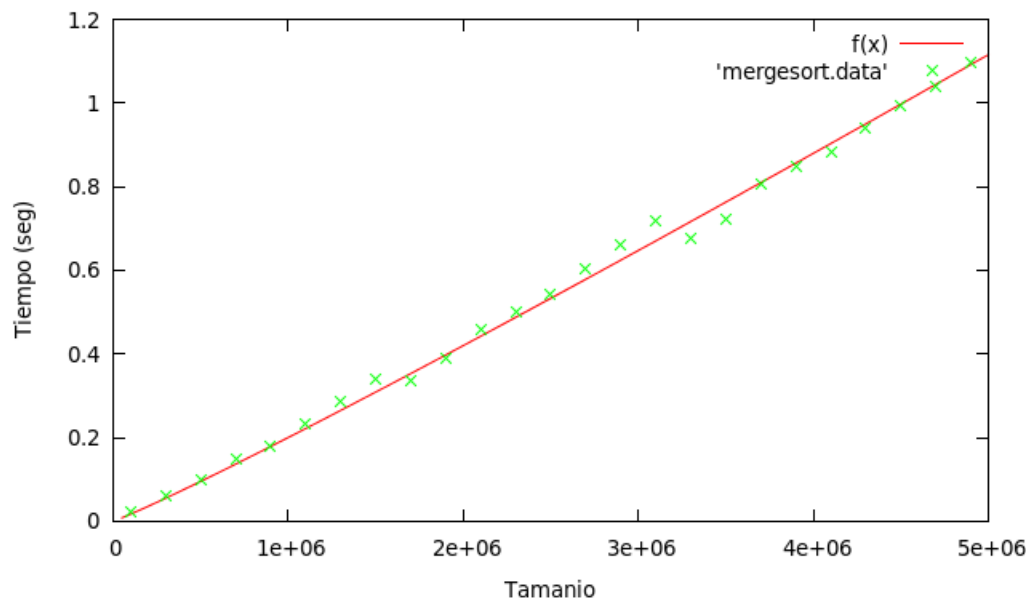
Constante	%error
2.44877e-09	0.3683%
8.77886e-10	0.2263%
1.08966e-09	0.2504%

3.5. Heapsort



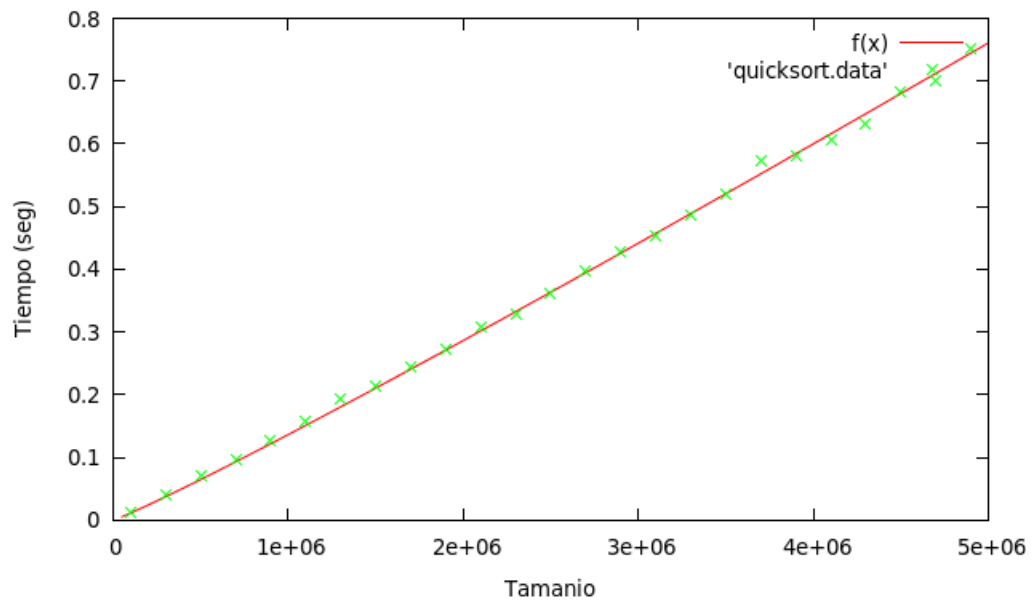
Constante	%error
1.70201e-08	0.6681%

3.6. Mergesort



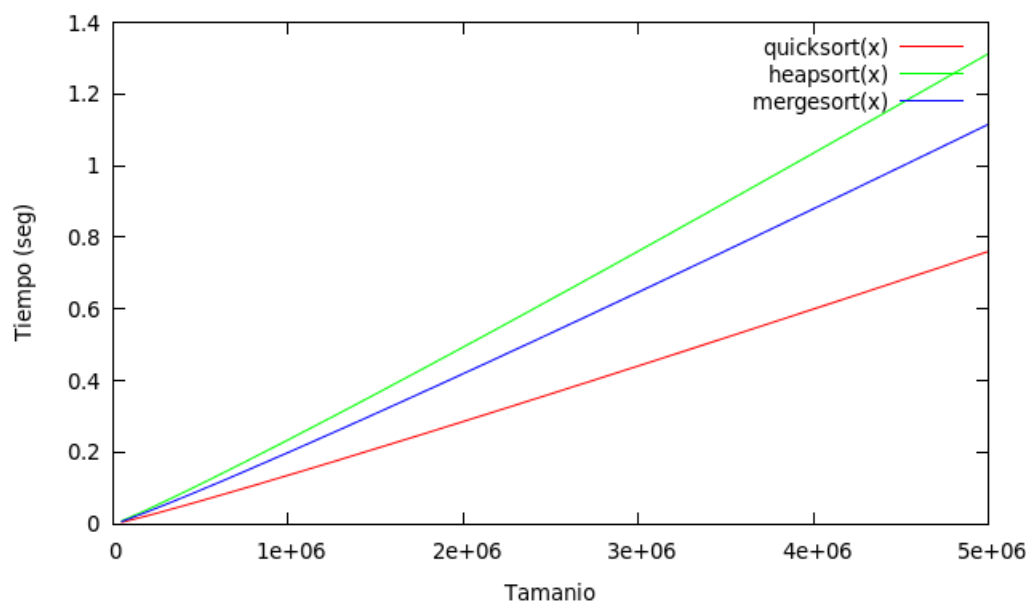
Constante	%error
1.44642e-08	0.6639%

3.7. Quicksort



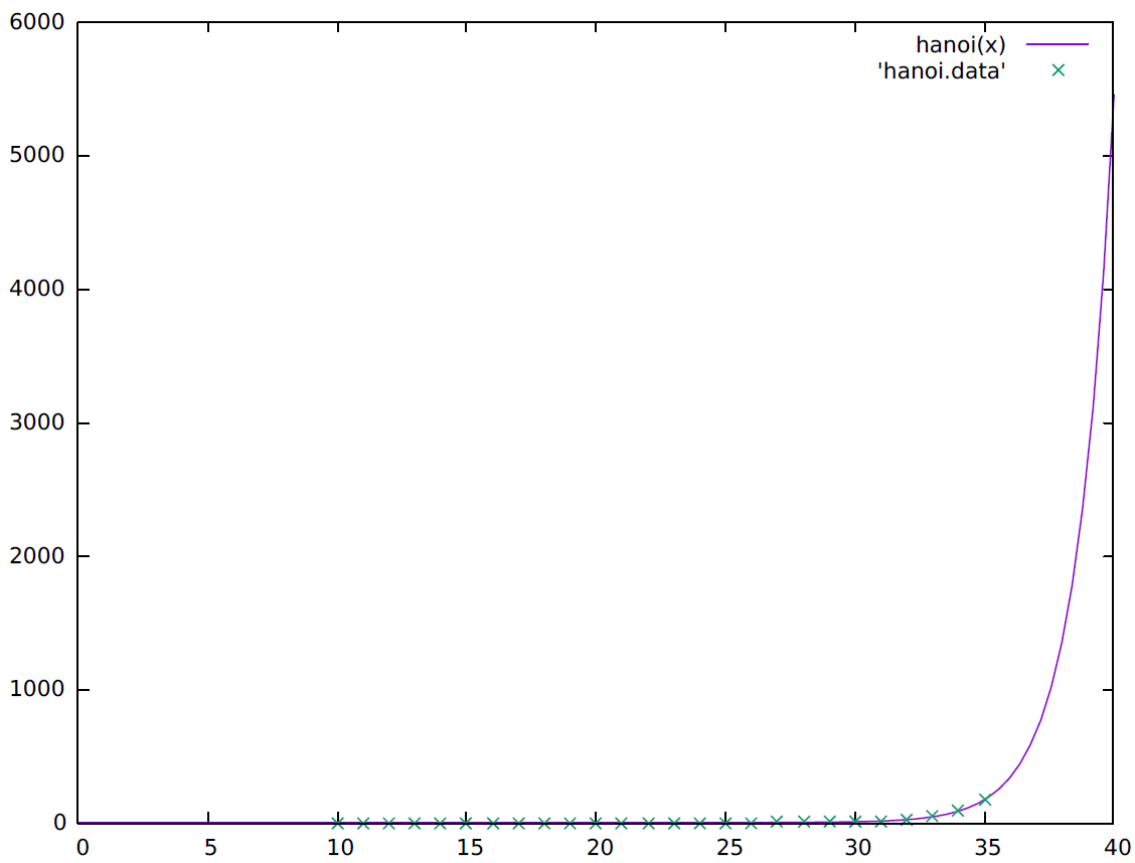
Constante	%error
9.86612e-09	0.3493%

3.8. Conjunto de algoritmos n^3



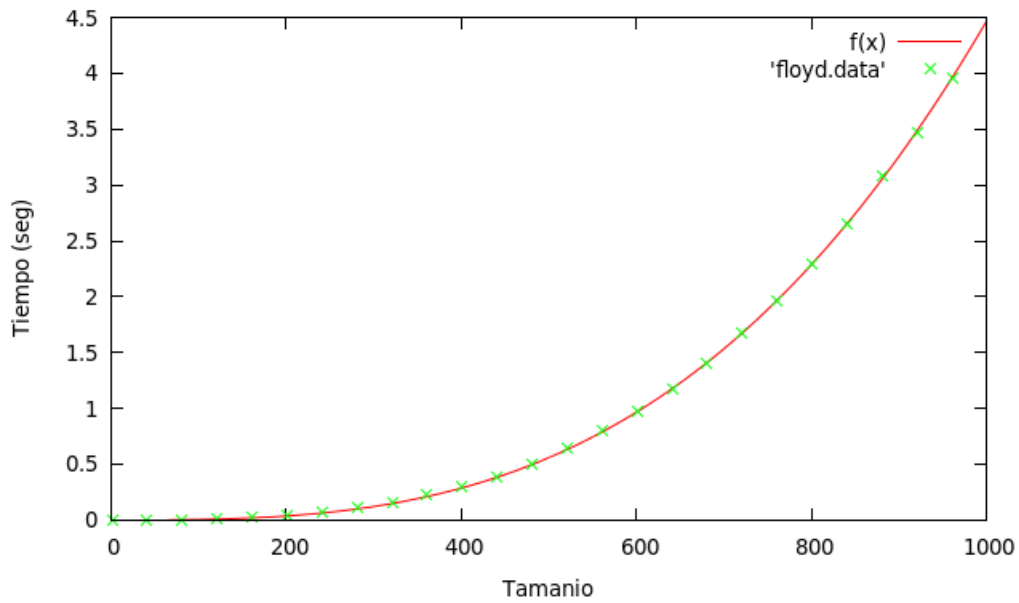
Constante	%error
9.86612e-09	0.3493%
1.44642e-08	0.6639%
1.70201e-08	0.6681%

3.9. Hanoi



Constante	%error
4.96044e-09	0.01422%

3.10. Floyd

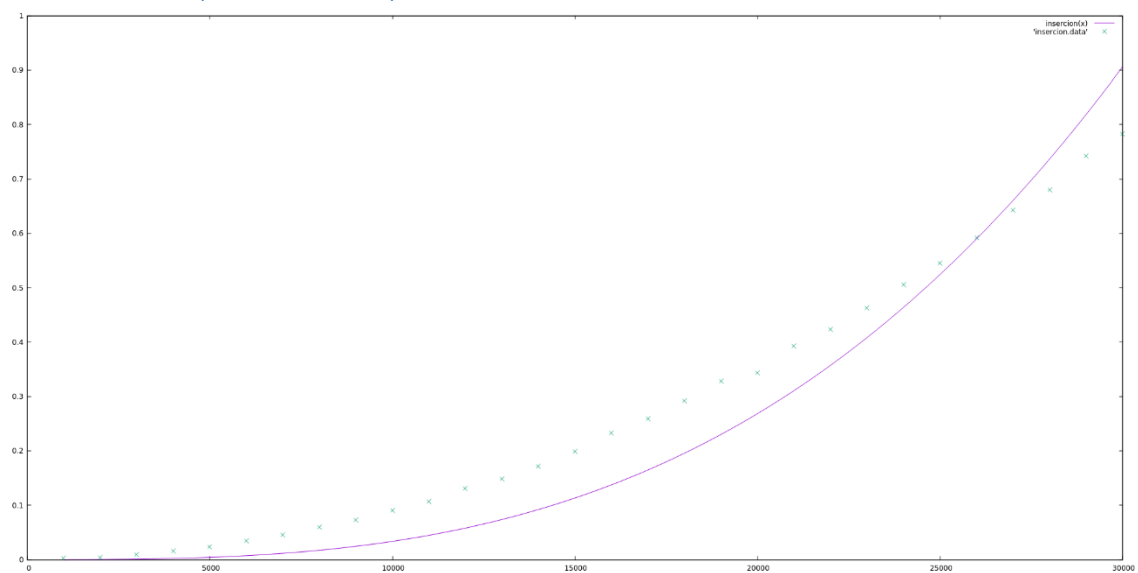


Constante	%error
4.46244e-09	0.1059%

4. Ajustes que no corresponden con la eficiencia teórica

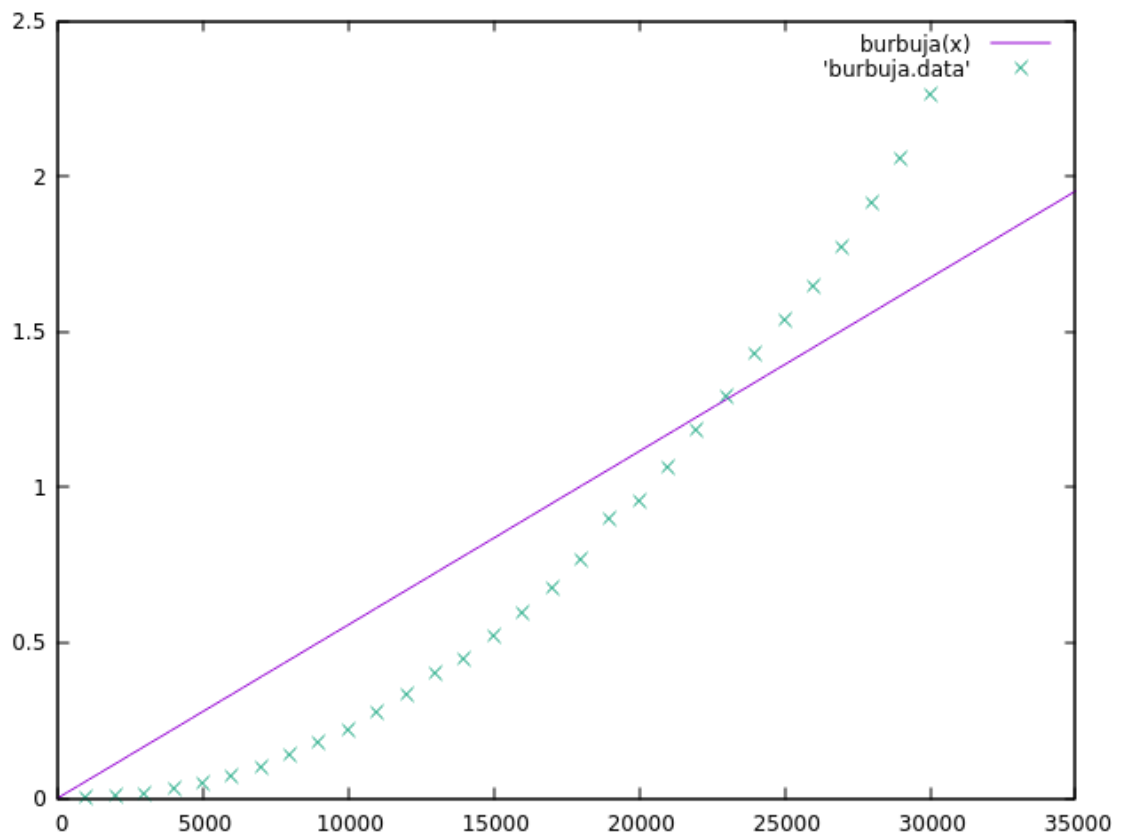
Hemos obtenido de los distintos algoritmos ajustes “malos” para obtener las gráficas y comprobar la variación de calidad del ajuste.

4.1. Inserción (orden cúbico)



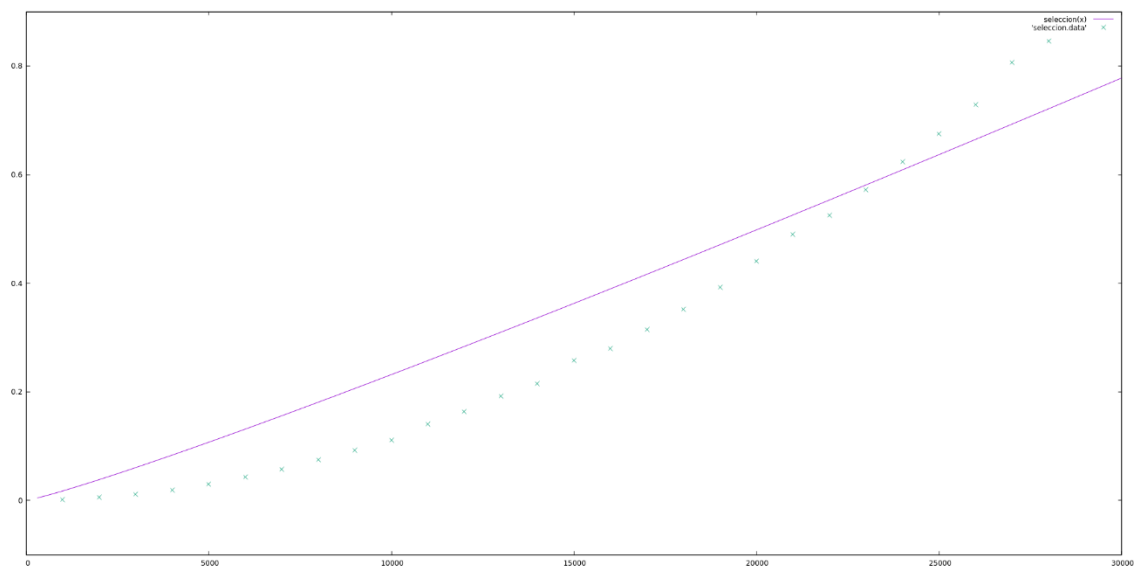
Constante	%error
3.35619e-14	3.263%

4.2. Burbuja (lineal)



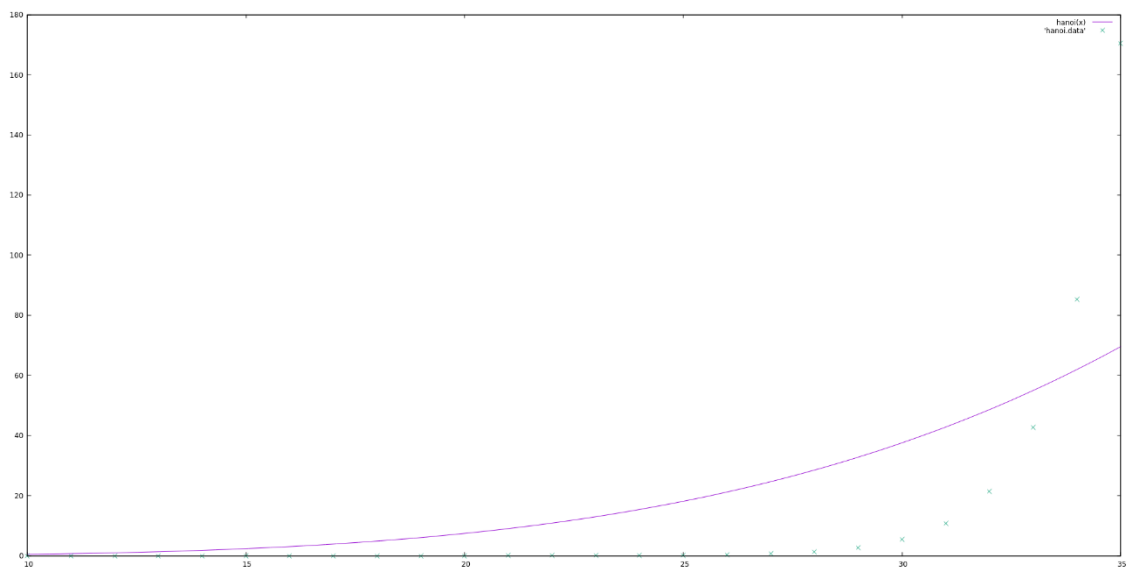
Constante	%error
5.57786e-05	5.102%

4.3. Selección (nlogn)



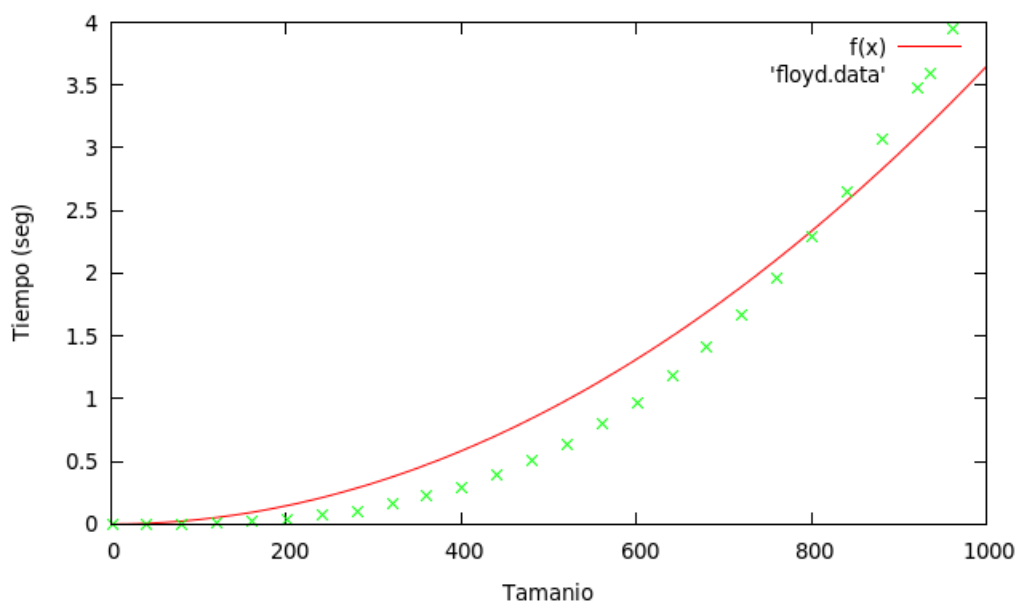
Constante	%error
2.51549e-06	4.043%

4.4. Hanoi (cuadrático)



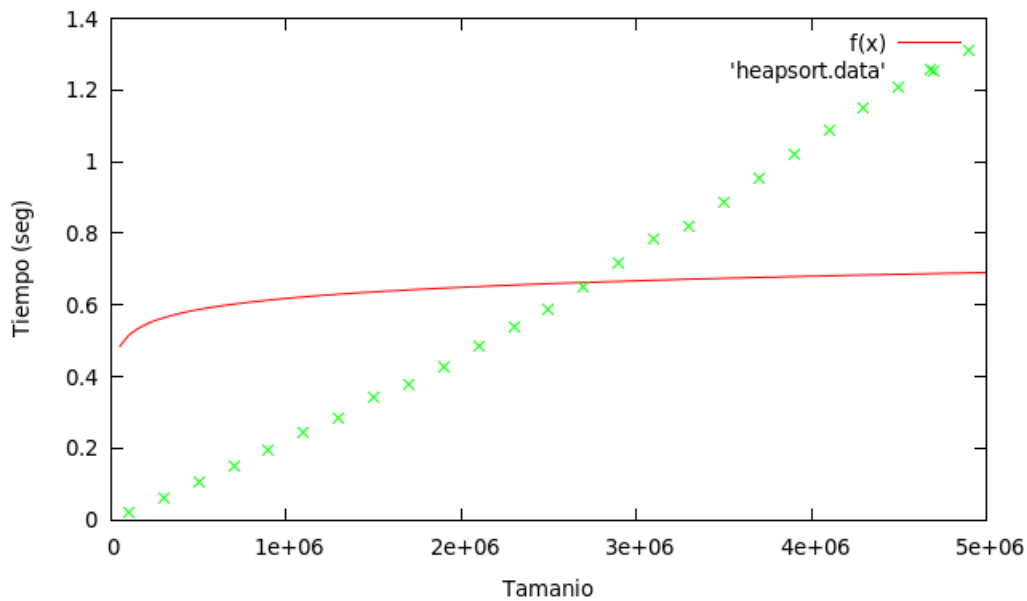
Constante	%error
4.63647e-05	18.06%

4.5. Floyd (cúbico)



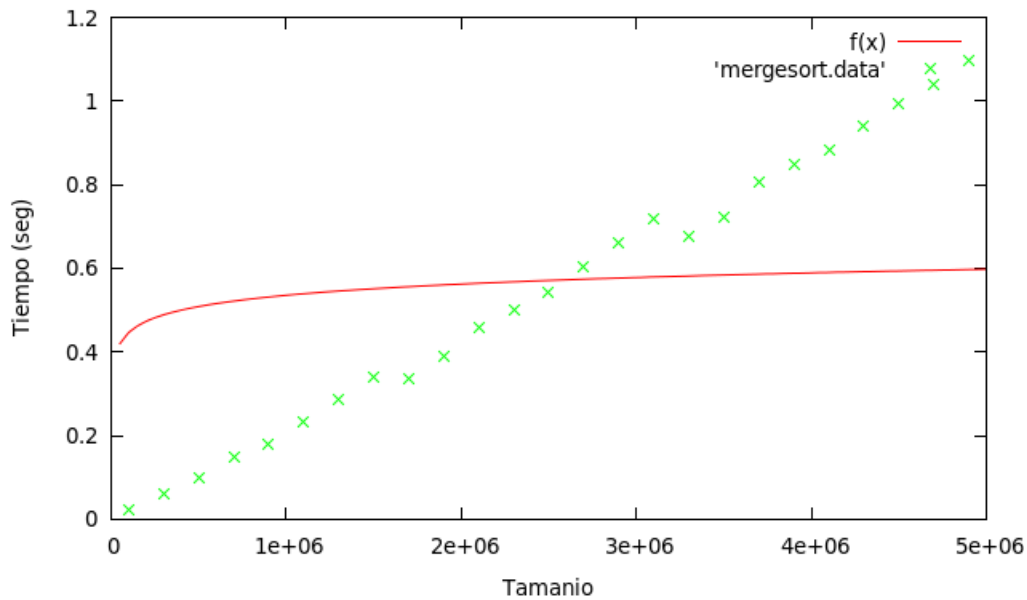
Constante	%error
3.64726e-06	3.383%

4.6. Heapsort (logn)



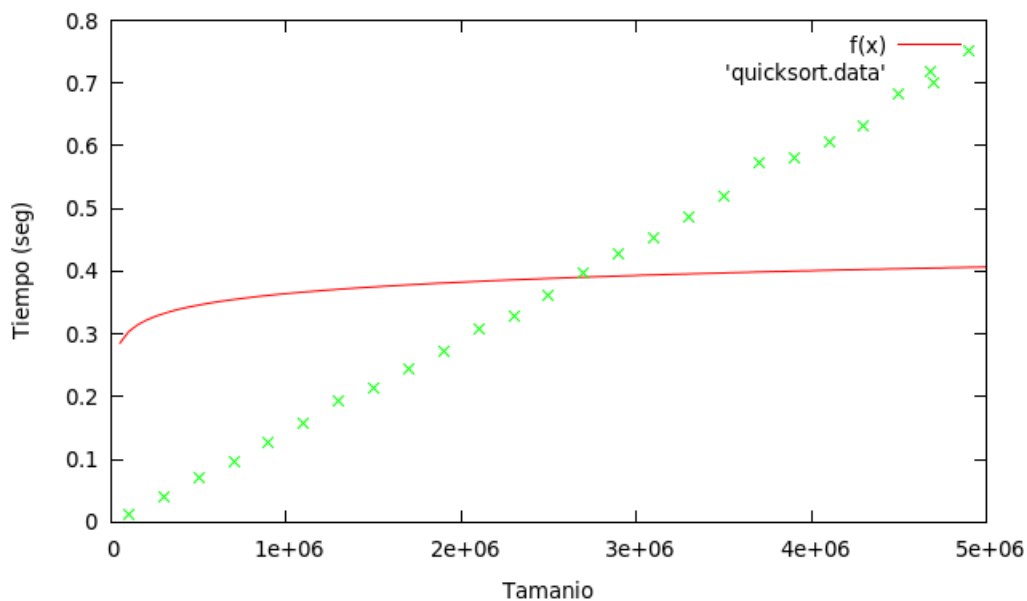
Constante	%error
0.0447823	11.38%

4.7. Mergesort(logn)



Constante	%error
0.0387524	10.5%

4.8. Quicksort (logn)



Constante	%error
0.0263752	10.59%

5. Datos empíricos en otro ordenador

Como última parte de la práctica, hemos obtenido los datos empíricos con los diferentes algoritmos en otro ordenador. Con los resultados obtenidos, podremos ver cómo no influye tanto el cambio de hardware en la eficiencia.

N	Hanoi	N	Floyd
10	0	1	0
11	0	41	0
12	0	81	0
13	0	121	0,015625
14	0	161	0,03125
15	0	201	0,0625
16	0	241	0,09375
17	0	281	0,140625
18	0	321	0,25
19	0	361	0,34375
20	0	401	0,40625
21	0,015625	441	0,546875
22	0,03125	481	0,703125
23	0,0625	521	0,921875
24	0,109375	561	1,14062
25	0,21875	601	1,375
26	0,46875	641	1,65625

27	0,875	681	2,01562
28	1,76562	721	2,54688
29	3,57812	761	3,15625
30	7	801	3,53125
31	14,0156	841	4,17188
32	27,9688	881	4,28125
33	56,7656	921	4,79688
34	111,766	961	5,34375
35	236,656		

N	Inserción	Selección	Burbuja
1000	0	0	0,015625
2000	0	0,015625	0,015625
3000	0	0,015625	0,015625
4000	0,015625	0,03125	0,03125
5000	0,03125	0,03125	0,046875
6000	0,03125	0,046875	0,09375
7000	0,046875	0,0625	0,125
8000	0,0625	0,09375	0,171875
9000	0,09375	0,109375	0,234375
10000	0,125	0,15625	0,28125
11000	0,15625	0,171875	0,34375
12000	0,171875	0,1875	0,40625
13000	0,1875	0,234375	0,484375
14000	0,21875	0,265625	0,5625
15000	0,25	0,296875	0,671875
16000	0,28125	0,34375	0,78125
17000	0,328125	0,390625	0,859375
18000	0,375	0,4375	0,96875
19000	0,453125	0,484375	1,07812
20000	0,453125	0,5625	1,20312
21000	0,5	0,625	1,34375
22000	0,546875	0,671875	1,5
23000	0,609375	0,734375	1,73438
24000	0,65625	0,78125	1,82812
25000	0,71875	0,859375	2,01562
26000	0,765625	0,921875	2,21875
27000	0,84375	1	2,32812
28000	0,890625	1,07812	2,5625
29000	0,96875	1,15625	2,60938
30000	1,03125	1,25	2,78125

N	Heapsort	Mergesort	Quicksort
100000	0,015625	0,015625	0,015625
300000	0,078125	0,078125	0,046875
500000	0,140625	0,125	0,078125
700000	0,171875	0,1875	0,109375
900000	0,234375	0,234375	0,140625
1100000	0,3125	0,3125	0,1875
1300000	0,375	0,375	0,21875
1500000	0,46875	0,453125	0,25
1700000	0,515625	0,53125	0,28125
1900000	0,609375	0,5625	0,328125
2100000	0,671875	0,609375	0,359375
2300000	0,765625	0,65625	0,390625
2500000	0,859375	0,734375	0,4375
2700000	0,875	0,78125	0,46875
2900000	0,953125	0,859375	0,53125
3100000	1,03125	0,9375	0,53125
3300000	1,10938	0,890625	0,59375
3500000	1,1875	0,953125	0,609375
3700000	1,26562	1,03125	0,640625
3900000	1,35938	1,09375	0,6875
4100000	1,42188	1,15625	0,734375
4300000	1,53125	1,21875	0,765625
4500000	1,625	1,34375	0,796875
4700000	1,73438	1,42188	0,84375
4900000	1,75	1,4375	0,875