

Práctica 1: Análisis de eficiencia de Algoritmos

Alejandro Sánchez Molina
Jorge García Moreno
David López Maldonado
Jose Manuel Rodríguez Calvo
Jesús Baeza Álvarez

Índice

1.-Introducción

2.-Cálculo de la eficiencia empírica

2.1.- Cálculo de los datos

2.2.-Tablas de datos de distinto orden.

- Algoritmos orden n^2 (Burbuja, inserción, selección)
- Algoritmos orden $n \log n$ (mergesort, heapsort, quicksort)
- Algoritmos orden n^3 (floyd)
- Algoritmos orden $((1 + \sqrt{5}/2)^n)$ (fibonacci)

3.-Gráficas comparativas de algoritmos

3.1.- Gráfica orden $O(n^2)$

3.2.- Gráfica orden $O(n \log n)$

3.3.- Gráfica orden $O(n^3)$

3.4.- Gráfica orden $((1 + \sqrt{5}/2)^n)$

3.5.- Gráfica de ordenación

4.- Cálculo eficiencia híbrida

5.- Comparación eficiencia empírica

5.1.- Prestaciones PCs utilizados

5.2.- Tiempos obtenidos para los distintos PCs

5.3.- Conclusiones de la comparación

Cálculo de los datos.

Para el cálculo de los datos hemos medido los recursos para cada tamaño dado las entradas. Para ello, partiendo de la macro proporcionada por el guión, hemos definido dos variables “tantes” y “tdespues” del tipo `clock_t` de la biblioteca `ctime` y a través de ellas hemos obtenido el tiempo según la entrada.

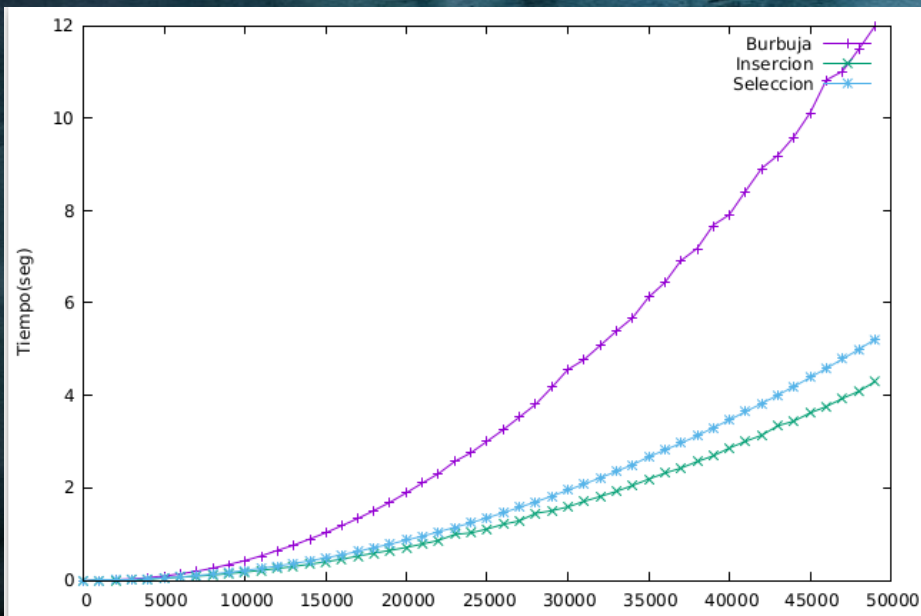
También hemos generado unas tablas y gráficos a partir de la salida de los ficheros de salida obtenidos a partir de los algoritmos.

Tabla de datos de distinto orden

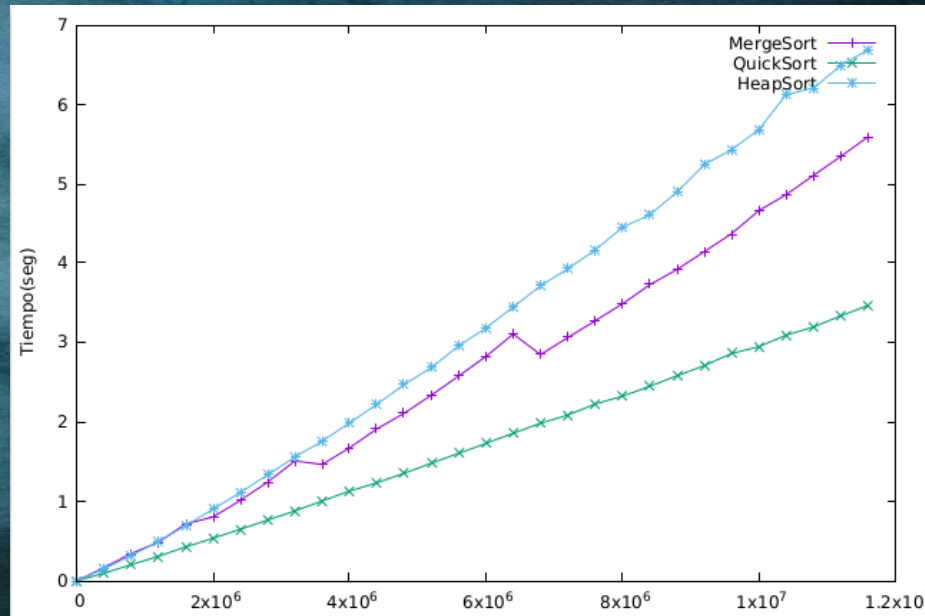
TAMAÑO	BURBUJA	INSERCIÓN	SELECCIÓN	TAMAÑO	MERGESORT	QUICKSORT	HEAPSORT	TAMAÑO	FLOYD	TAMAÑO	FIBONACCI
1000	0.003474	0.002273	0.002269	400000	0.166124	0.097076	0.15	0	0.000338	21	0.000124
2000	0.013642	0.007475	0.008776	800000	0.345536	0.205298	0.319394	32	0.000338	22	0.0002
3000	0.030865	0.01713	0.019746	1200000	0.487227	0.309781	0.494658	64	0.002549	23	0.000344
4000	0.06077	0.030512	0.034904	1600000	0.719383	0.429227	0.69876	96	0.008875	24	0.00052
5000	0.094498	0.051514	0.054466	2000000	0.806097	0.538046	0.90567	128	0.020695	25	0.00084
6000	0.143243	0.070034	0.078158	2400000	1.01498	0.651421	1.11594	160	0.038876	26	0.001629
7000	0.198503	0.093721	0.105868	2800000	1.24496	0.765841	1.33647	192	0.066284	27	0.002291
8000	0.266597	0.116877	0.137996	3200000	1.5106	0.883142	1.56027	224	0.104604	28	0.00357
9000	0.344918	0.148621	0.174752	3600000	1.46806	1.00774	1.7613	256	0.15652	29	0.005771
10000	0.42949	0.182247	0.215889	4000000	1.67631	1.13079	1.99007	288	0.219608	30	0.009327
11000	0.530335	0.217741	0.260781	4400000	1.91485	1.23534	2.22281	320	0.298932	31	0.016051
12000	0.648409	0.257772	0.311337	4800000	2.11166	1.35673	2.4737	352	0.399858	32	0.024939
13000	0.762505	0.301912	0.366014	5200000	2.34108	1.48331	2.68743	384	0.519754	33	0.039914
14000	0.895291	0.352296	0.424018	5600000	2.58193	1.60899	2.96379	416	0.661122	34	0.064544
15000	1.03	0.402478	0.48658	6000000	2.82843	1.73237	3.18594	448	0.821598	35	0.103813
16000	1.18292	0.458153	0.554026	6400000	3.10898	1.85956	3.45199	480	1.01143	36	0.167294
17000	1.34286	0.52831	0.625542	6800000	2.85215	1.98569	3.71626	512	1.22582	37	0.269919
18000	1.51497	0.592809	0.715723	7200000	3.06476	2.0898	3.93041	544	1.47021	38	0.43716
19000	1.69277	0.650443	0.783128	7600000	3.27799	2.22835	4.16972	576	1.74266	39	0.706828
20000	1.89772	0.716637	0.874388	8000000	3.49011	2.32794	4.45239	608	2.05254	40	1.14355
21000	2.11058	0.790636	0.956913	8400000	3.73093	2.45234	4.61333	640	2.38865	41	1.84923
22000	2.31255	0.866258	1.05346	8800000	3.92282	2.58512	4.89892	672	2.77026	42	2.99506
23000	2.57535	0.995187	1.14626	9200000	4.14908	2.71357	5.24467	704	3.18981	43	4.84251
24000	2.76332	1.02884	1.2481	9600000	4.37556	2.86652	5.43264	736	3.18981	44	7.92531
25000	3.00934	1.11482	1.3532	10000000	4.66431	2.95001	5.6845	768	4.14594	45	12.6945

Gráficas Comparativas

-Orden(n^2)

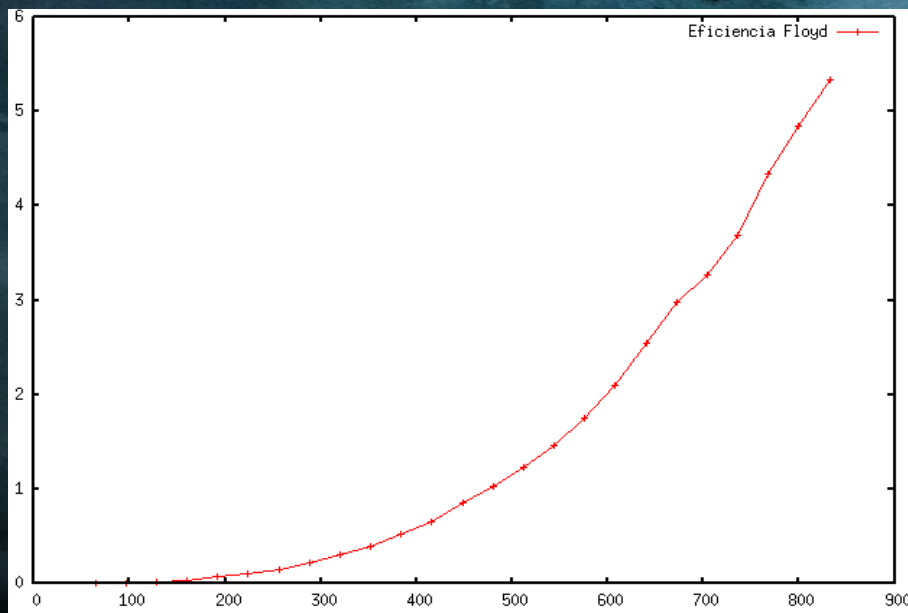


-Orden($n \cdot \log(n)$)

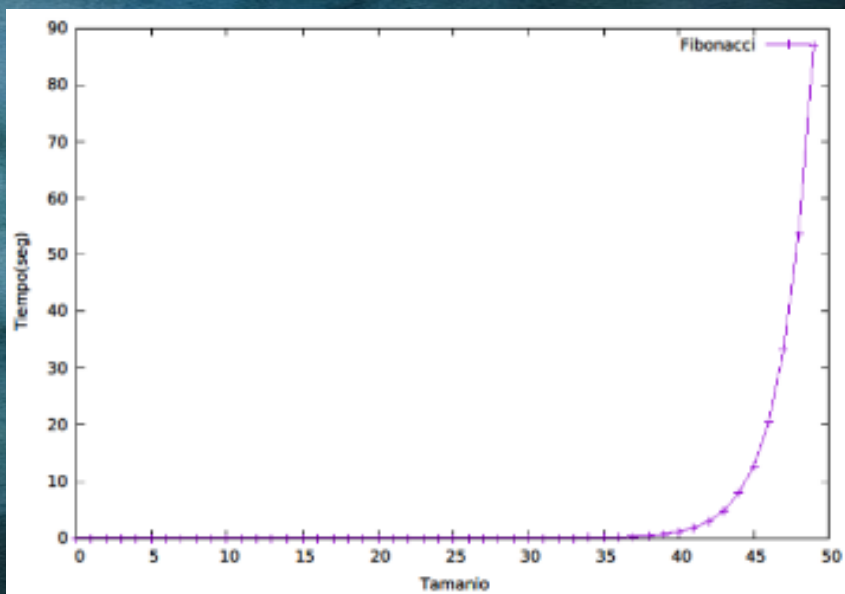


Gráficas Comparativas

-Orden n^3



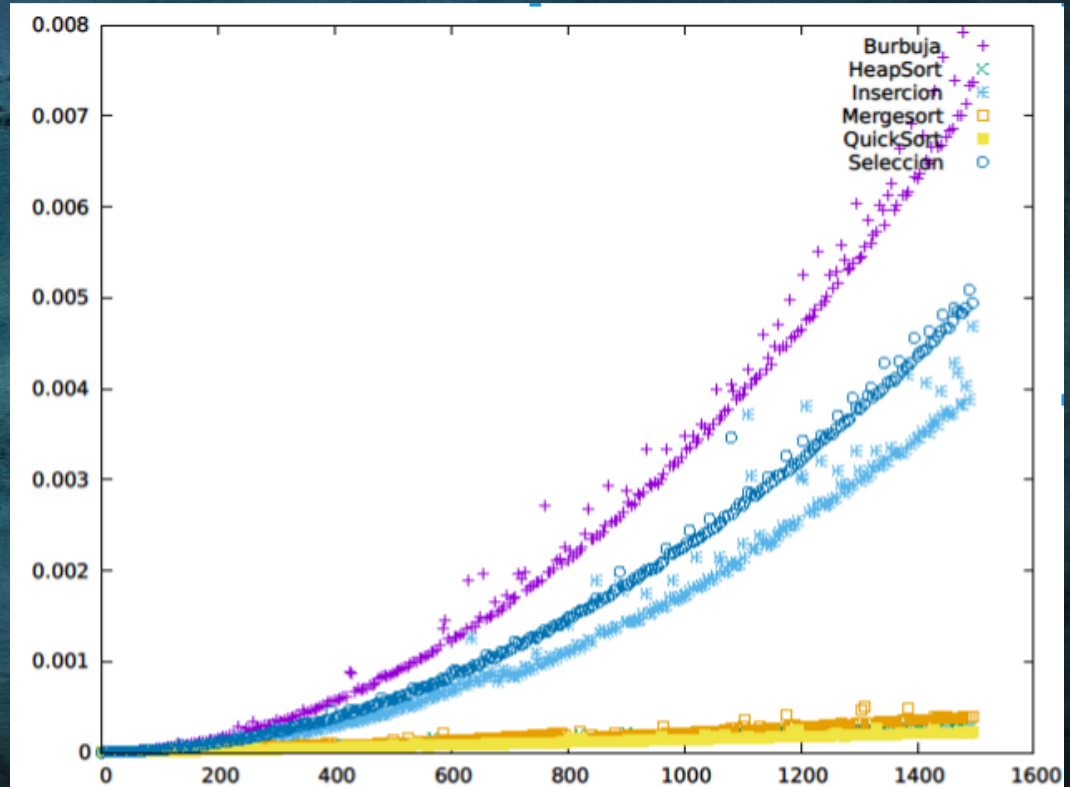
-Orden $1,618^n$



Gráficas Comparativas

-Gráfica con algoritmos de
ordenación

n^2 y $n \cdot \log(n)$



Eficiencia Híbrida.

Para cada grupo de algoritmos, hemos calculado la eficiencia híbrida en base a su fórmula. Primero, en el caso de los algoritmos de Inserción, Burbuja y Selección, la fórmula que hemos empleado es: $f(x) = a*x^2+b*x+c$.

Donde en el programa Gnuplot, que es el que hemos empleado para obtener este cálculo y para obtener las tablas es: $f(x) = a0*x*x+a1*x+a2$.

Inserción:

a (a0) = 1.79012e-09 +/- 4.433e-12 (0.2476%)
b (a1) = -2.75067e-07 +/- 2.286e-07 (83.12%)
c (a2) = 0.00259602 +/- 0.002478 (95.45%)

Burbuja:

a (a0) = 3.74848e-09 +/- 1.852e-11 (0.4939%)
b (a1) = -7.29409e-06 +/- 9.382e-07 (12.86%)
c (a2) = 0.00878894 +/- 0.00994 (113.1%)

En las cuales, el margen de error obtenido aparece a la derecha.

Selección:

a (a0) = 2.21783e-09	+/- 1.115e-11	(0.5027%)
b (a1) = -1.70046e-06	+/- 5.404e-07	(31.78%)
c (a2) = 0.00922901	+/- 0.005506	(59.66%)

Para los Algoritmos Mergesort, quicksort y Heapsort, la función que hemos calculado es: $f(x) = a \cdot x \cdot \log(x)$, donde en el programa Gnuplot es:

$f(x) = a0 + x + \log(x)$. Y los valores de los factores multiplicativos y del margen de error para cada algoritmo son:

Mergesort.

$a = 3.25199e-08 \pm 4.022e-10$ (1.237%)

Quicksort.

$a = 1.98491e-08 \pm 1.639e-10$ (0.8258%)

Heapsort.

$a = 2.83396e-08 \pm 1.941e-10$ (0.6849%)

Floyd.

Para Floyd, la función calculada es: $f(x) = a*x^3+b*x^2+c*x+d$.

Donde en el Gnuplot la función es: $f(x) = a0*x*x*x+a1*x*x+c*x+d$.

$a = 4.5651e-14$	$\pm 3.738e-15$	(8.188%)
$b = -1.52907e-12$	$\pm 1.217e-13$	(7.961%)
$c = 32$	$\pm 1.204e-12$	(3.762e-12%)
$d = -1.35692e-11$	$\pm 3.806e-12$	(28.05%)

Fibonacci.

Para el Algoritmo de Fibonacci, hemos utilizado la aproximación dada en la práctica, de manera que la fórmula que hemos empleado para el cálculo de los datos ha sido: $f(x) = a \cdot 1,618^x$. Siendo la fórmula en el programa Gnuplot:

$f(x) = 1.618^{(a0 \cdot x)}$. Donde el valor del factor multiplicativo y del margen de error es:

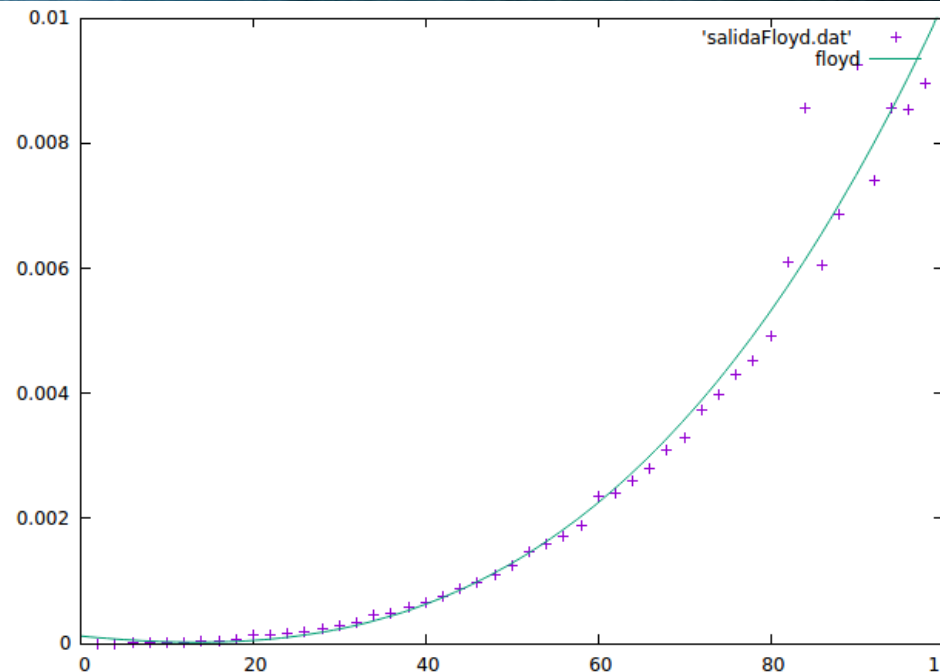
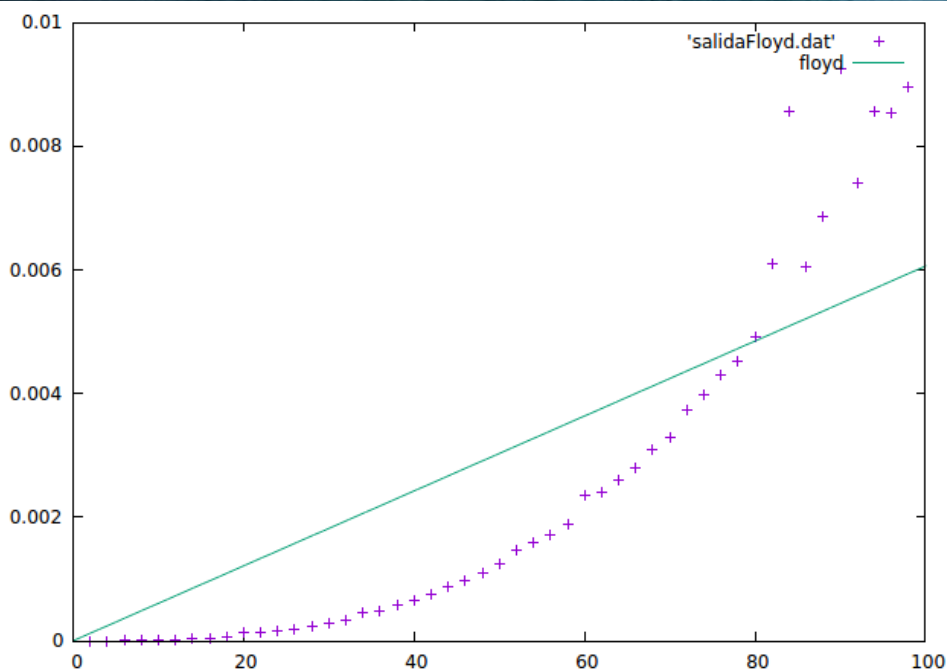
$$a = 5.03885e-09 \quad \pm 2.085e-12 \quad (0.04137\%)$$

Floyd con $f(x)=a*x$

$a= 6.05878e-05$ $\pm 4.002e-06$ (6.606%)

Ajuste erróneo

Ajuste bueno

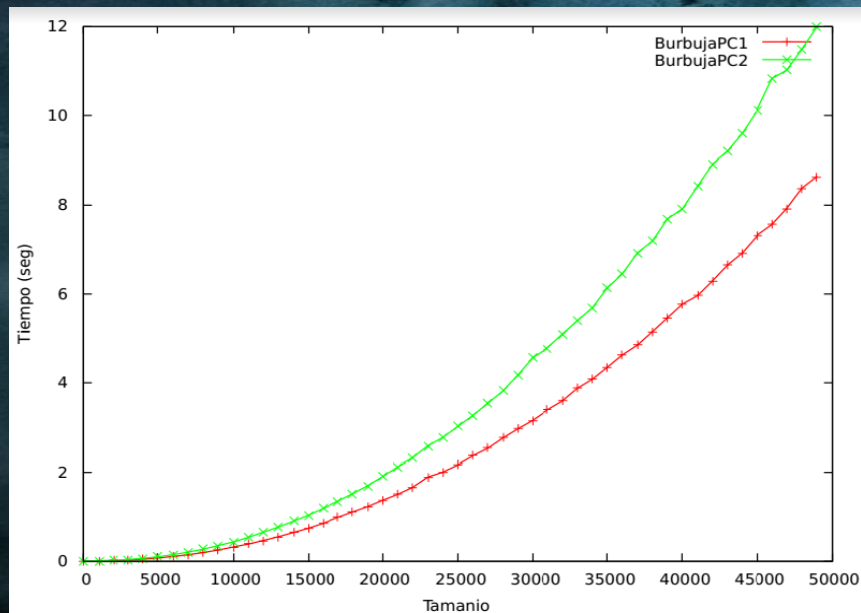


Prestaciones de PCs utilizados

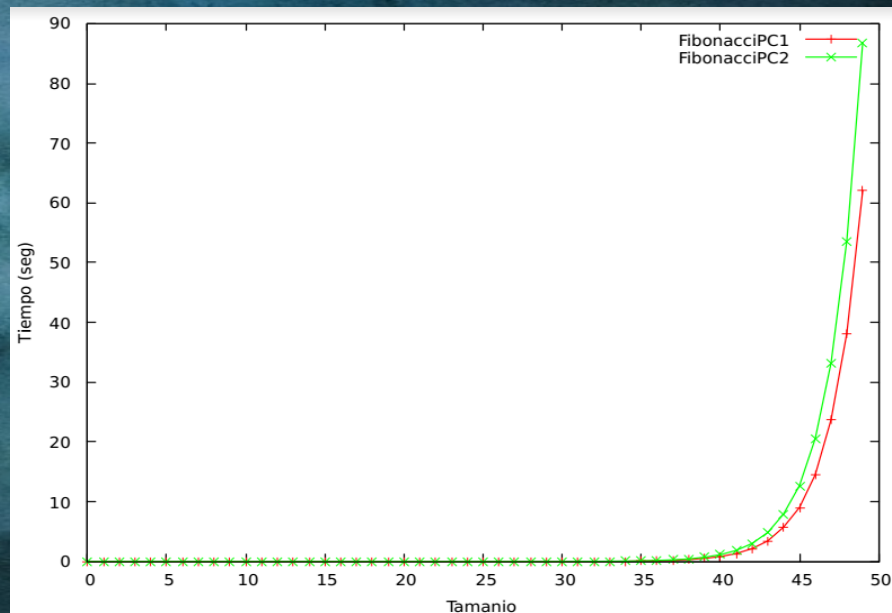
Para tomar medidas hemos utilizado 2 ordenadores distintos, uno está equipado con un procesador i5 a 2.2 GHz y otro un i3 a 2.0GHz, para hacer esta comparativa no teníamos ningún otro proceso abierto.



Tiempos obtenidos para los distintos PCs (Comparativa de gráficas) I

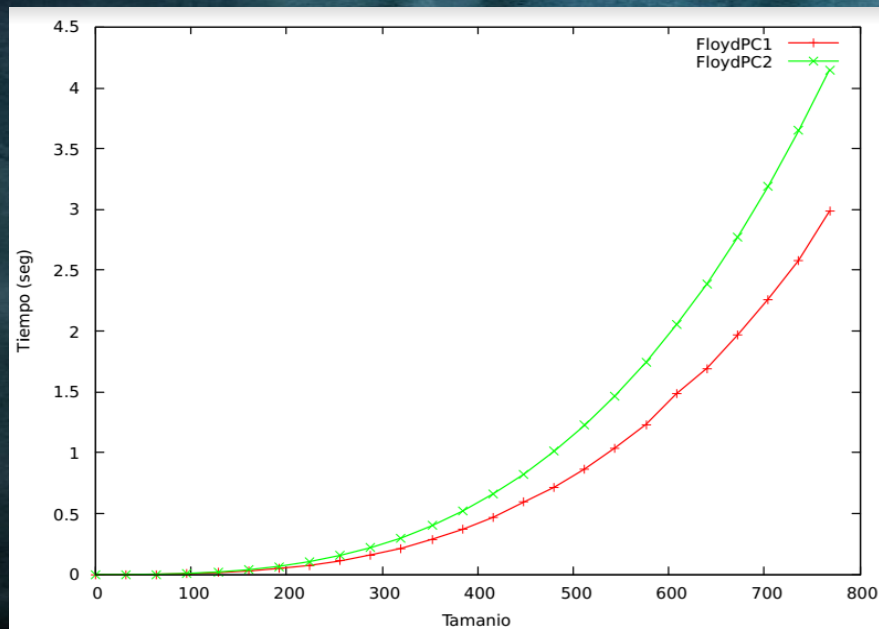


$O(n^2)$

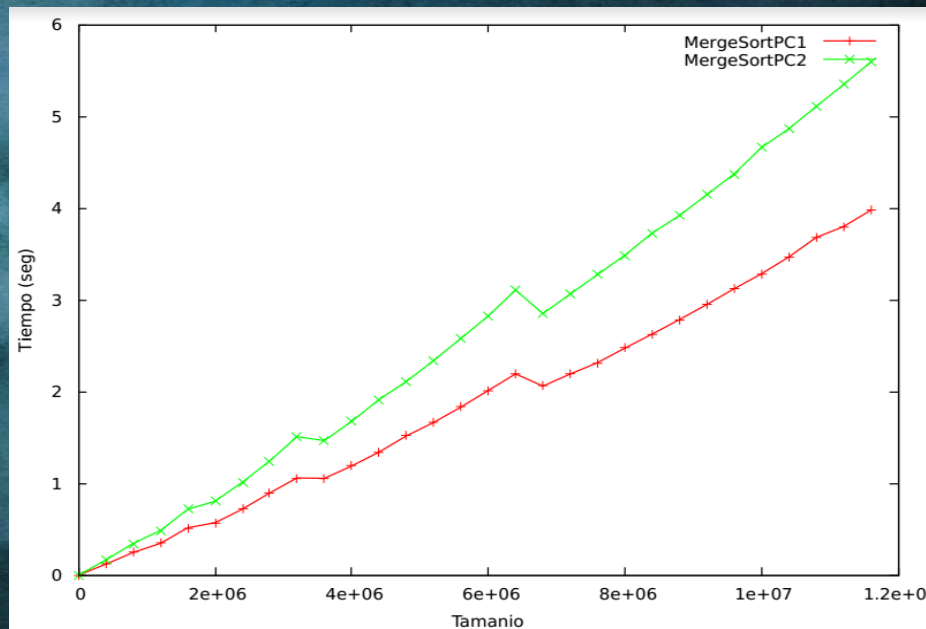


$O(1,618^n)$

Tiempos obtenidos para los distintos PCs (Comparativa de gráficas) II



$O(n^3)$



$O(n \log n)$

Conclusiones de la comparación

Como podemos observar en las gráficas el tiempo de ejecución de los algoritmos con un Intel Core i5-5200U 2.2GHz son menores que los algoritmos ejecutados en el Intel Core i3-5005U 2.0GHz . Esto es debido diferentes factores internos que hacen que el i5 supere al i3.