

# Algorítmica: Práctica 1

*Andrés Herrera Poyatos, Antonio Rafael Moya Martín-Castaño, Iván Sevillano García, Juan Luis Suárez Díaz*

*15/3/2015*

## Contents

<b>Organización de la práctica</b>	<b>2</b>
<b>Ejercicio 1: Cálculo de la eficiencia empírica</b>	<b>2</b>
Tabla con los algoritmos cuadráticos . . . . .	2
Tabla con los algoritmos cúbicos . . . . .	3
Tabla con el algoritmo de Fibonacci ( $O((\frac{1+\sqrt{5}}{2})^n)$ ) . . . . .	4
Tabla con el algoritmo de Hanoi ( $O(2^n)$ ) . . . . .	5
Tabla con los algoritmos $n \log n$ . . . . .	5
Tabla con los algoritmos de ordenación . . . . .	6
<b>Ejercicio 2: Elaboración de gráficas.</b>	<b>8</b>
Gráfica comparativa de los algoritmos cuadráticos. . . . .	8
Gráfica del algoritmo cúbico (Floyd) . . . . .	8
Gráfica del algoritmo de Fibonacci ( $O((\frac{1+\sqrt{5}}{2})^n)$ ) . . . . .	9
Gráfica del algoritmo de Hanoi ( $O(2^n)$ ) . . . . .	9
Gráfica de los algoritmos $O(n \log n)$ . . . . .	10
Gráfica comparativa con todos los algoritmos de ordenación. . . . .	10

## Organización de la práctica

Se adjunta el directorio comprimido **code** con todos los datos obtenidos. La información se organiza como sigue:

- Los códigos **.cpp** de los distintos algoritmos están disponibles en la carpeta **src**.
- Se añade el script de bash **ejecuciones.sh**, el cual se encarga de obtener todos los datos y gráficas para todos los algoritmos.
- En la carpeta **sh** se añaden scripts auxiliares, cada uno especializado en la toma de datos de uno o varios algoritmos concretos.
- En la carpeta **plot**, de la misma forma, se encuentran scripts especializados en la elaboración de las distintas gráficas.
- En las carpetas **DatosAutor** se almacenan los archivos **.dat** generados por cada uno de los autores, en sus respectivos PCs. Los ficheros contienen, para cada algoritmo, varias parejas *[tamaño tiempo]* correspondientes a distintas ejecuciones del programa con distintos tamaños y sus respectivos resultados, con los datos utilizados en el trabajo.
- De forma análoga, están disponibles los directorios **TablasAutor** e **ImagenesAutor** que contienen tablas en formato **.md** con los resultados, y las gráficas del comportamiento de los algoritmos generadas por *gnuplot*, respectivamente.

Cada ejercicio tiene su apartado en el pdf con su correspondiente enunciado y solución.

**Nota:** Se ha utilizado el shell bash para la obtención de los distintos datos de los algoritmos.

## Ejercicio 1: Cálculo de la eficiencia empírica

Calcule la eficiencia empírica de los algoritmos pedidos. Defina adecuadamente los tamaños de entrada para 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  $O(n^2)$ , otra con los  $O(n \log n)$ , otra con  $O(n^3)$  y otra con  $O((\frac{1+\sqrt{5}}{2})^n)$ ).

A continuación se disponen las distintas tablas para cada clase de algoritmos:

### Tabla con los algoritmos cuadráticos

Tamaño del Vector	Burbuja	Seleccion	Insercion
1000	0.005971	0.003397	0.001321
2000	0.018136	0.009589	0.007588
3000	0.024143	0.014704	0.020282
4000	0.043267	0.025817	0.023064
5000	0.067684	0.037817	0.034221
6000	0.099499	0.055028	0.047872
7000	0.137072	0.073739	0.064517
8000	0.181558	0.092111	0.082905
9000	0.232648	0.118624	0.103043
10000	0.290489	0.14394	0.124546

Tamaño del Vector	Burbuja	Seleccion	Insercion
11000	0.354349	0.178614	0.151216
12000	0.433737	0.20678	0.178228
13000	0.519202	0.239558	0.209278
14000	0.59308	0.273397	0.248141
15000	0.689312	0.314147	0.276967
16000	0.789129	0.356495	0.317291
17000	0.890449	0.402106	0.358508
18000	1.01538	0.450575	0.397242
19000	1.1313	0.50472	0.435913
20000	1.26128	0.55525	0.483853
21000	1.39441	0.611367	0.541654
22000	1.55788	0.670662	0.600085
23000	1.68169	0.732809	0.644882
24000	1.84769	0.800821	0.70273
25000	1.9893	0.864984	0.762199

**Tabla con los algoritmos cúbicos**

Nodos del Grafo	Floyd
32	0.000596
64	0.004593
96	0.01017
128	0.017141
160	0.035407
192	0.054113
224	0.083649
256	0.116013
288	0.153556
320	0.217792
352	0.280357
384	0.362685
416	0.460287
448	0.581175
480	0.703839
512	0.852424
544	1.02124

Nodos del Grafo	Floyd
576	1.25977
608	1.44669
640	1.68365
672	1.93344
704	2.23303
736	2.54158
768	2.89293
800	3.25971

**Tabla con el algoritmo de Fibonacci ( $O((\frac{1+\sqrt{5}}{2})^n)$ )**

Índice	Fibonacci
15	1.3e-05
16	2e-05
17	2.6e-05
18	4.4e-05
19	5e-05
20	0.000114
21	8.6e-05
22	0.000154
23	0.000582
24	0.00097
25	0.001314
26	0.002554
27	0.002394
28	0.003356
29	0.004289
30	0.007083
31	0.011583
32	0.017354
33	0.029313
34	0.047371
35	0.073093
36	0.127835
37	0.190808
38	0.308124

Índice	Fibonacci
39	0.498824
40	0.849934

Tabla con el algoritmo de Hanoi ( $O(2^n)$ )

Num. Discos	Hanoi
5	1e-06
6	3e-06
7	3e-06
8	6e-06
9	9e-06
10	1.3e-05
11	4.9e-05
12	7.6e-05
13	0.00015
14	0.00019
15	0.000393
16	0.000851
17	0.002302
18	0.003382
19	0.009191
20	0.019015
21	0.024593
22	0.041194
23	0.065421
24	0.127555
25	0.246427
26	0.483075
27	0.96832
28	1.9249
29	3.83247
30	7.63996

Tabla con los algoritmos  $n \log n$

Tamaño del Vector	Mergesort	Quicksort
40000	0.015087	0.006235
80000	0.02682	0.014736
120000	0.037756	0.02246
160000	0.041266	0.025439
200000	0.059359	0.032775
240000	0.057706	0.041055
280000	0.065938	0.045861
320000	0.082393	0.053183
360000	0.093771	0.057395
400000	0.107337	0.063843
440000	0.102685	0.071064
480000	0.122825	0.076521
520000	0.136037	0.082585
560000	0.141045	0.087434
600000	0.150005	0.093448
640000	0.1658	0.100634
680000	0.181068	0.109131
720000	0.211107	0.115456
760000	0.205422	0.121493
800000	0.226734	0.129283
840000	0.200972	0.136155
880000	0.211482	0.141553
920000	0.23571	0.148845
960000	0.240497	0.155352
1000000	0.244299	0.178312

Finalmente, mostramos una tabla con la comparativa de todos los algoritmos de ordenación, tanto cuadráticos como  $n \log n$ . Podemos apreciar, que para tamaños relativamente pequeños (25.000), ya se muestran diferencias bastante grandes:

### Tabla con los algoritmos de ordenación

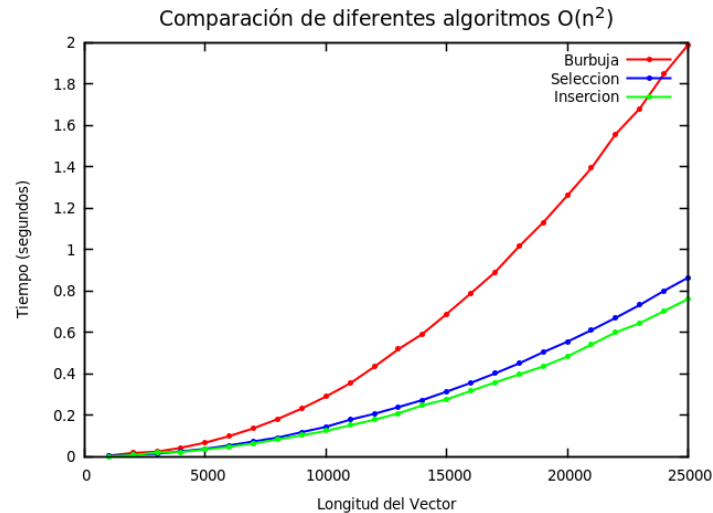
Tamaño del Vector	Burbuja	Selecccion	Insercion	Mergesort	Quicksort
1000	0.005971	0.003397	0.001321	0.000359	0.000195
2000	0.018136	0.009589	0.007588	0.000756	0.000269
3000	0.024143	0.014704	0.020282	0.00074	0.000671
4000	0.043267	0.025817	0.023064	0.000748	0.00067

Tamaño del Vector	Burbuja	Seleccion	Insercion	Mergesort	Quicksort
5000	0.067684	0.037817	0.034221	0.002255	0.000567
6000	0.099499	0.055028	0.047872	0.001549	0.001661
7000	0.137072	0.073739	0.064517	0.003041	0.001953
8000	0.181558	0.092111	0.082905	0.003166	0.002456
9000	0.232648	0.118624	0.103043	0.004058	0.001649
10000	0.290489	0.14394	0.124546	0.003803	0.001971
11000	0.354349	0.178614	0.151216	0.004144	0.002048
12000	0.433737	0.20678	0.178228	0.0041	0.003616
13000	0.519202	0.239558	0.209278	0.005279	0.003037
14000	0.59308	0.273397	0.248141	0.006677	0.002399
15000	0.689312	0.314147	0.276967	0.006024	0.002247
16000	0.789129	0.356495	0.317291	0.007461	0.002721
17000	0.890449	0.402106	0.358508	0.006324	0.002745
18000	1.01538	0.450575	0.397242	0.008806	0.005765
19000	1.1313	0.50472	0.435913	0.008887	0.004034
20000	1.26128	0.55525	0.483853	0.007868	0.003407
21000	1.39441	0.611367	0.541654	0.00544	0.00359
22000	1.55788	0.670662	0.600085	0.006238	0.00374
23000	1.68169	0.732809	0.644882	0.010339	0.006076
24000	1.84769	0.800821	0.70273	0.010512	0.006921
25000	1.9893	0.864984	0.762199	0.009398	0.003104

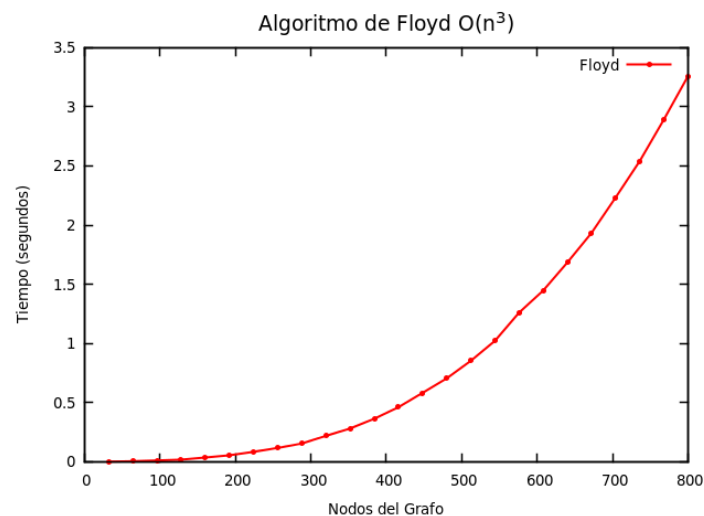
## Ejercicio 2: Elaboración de gráficas.

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 de rendimiento de algoritmos con diferente orden de eficiencia.

### Gráfica comparativa de los algoritmos cuadráticos.

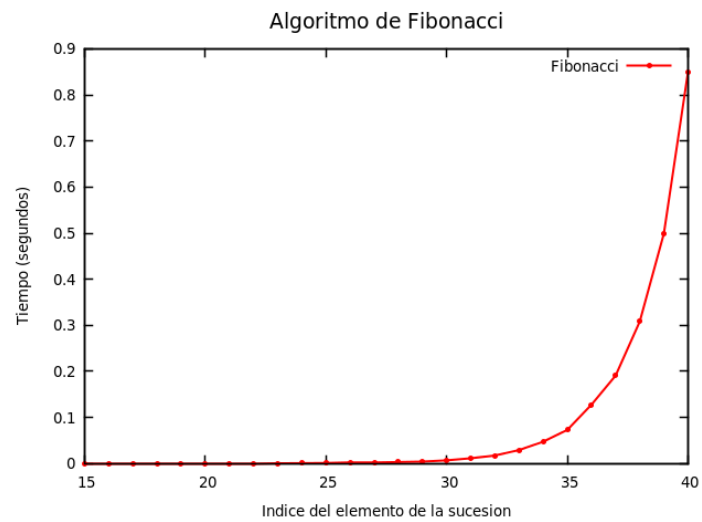


### Gráfica del algoritmo cúbico (Floyd)

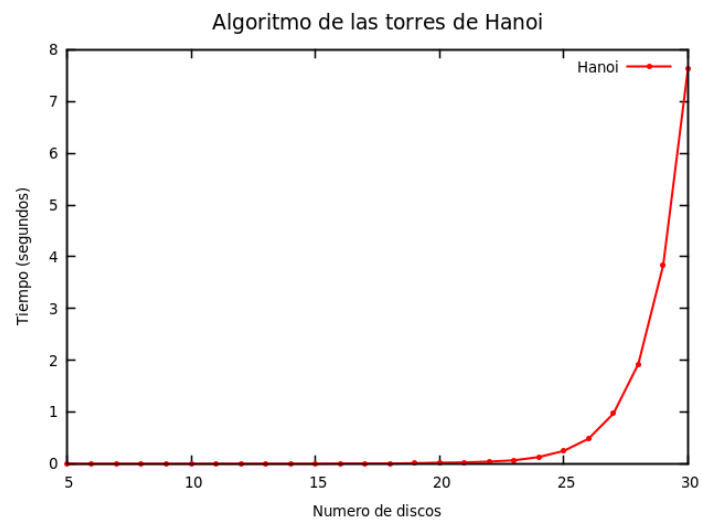




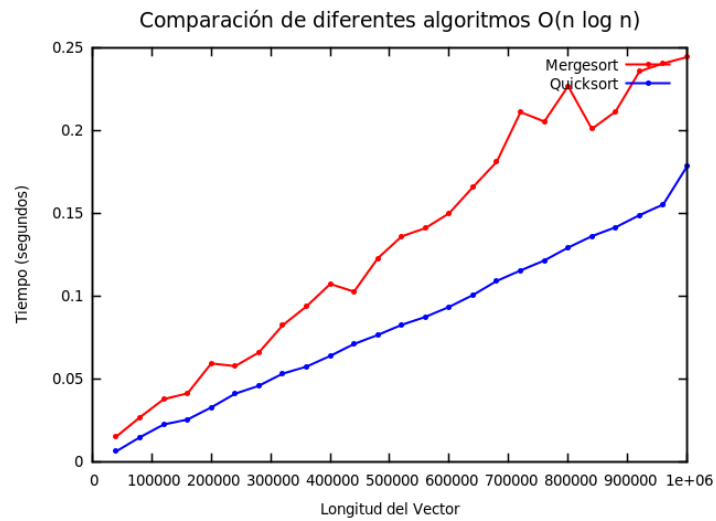
### Gráfica del algoritmo de Fibonacci ( $O((\frac{1+\sqrt{5}}{2})^n)$ )



### Gráfica del algoritmo de Hanoi ( $O(2^n)$ )



## Gráfica de los algoritmos $O(n \log n)$



## Gráfica comparativa con todos los algoritmos de ordenación.

