

ANÁLISIS DE EFICIENCIA DE ALGORITMOS





¡HOLA!

Somos el Grupo 3

Gregorio Carvajal Expósito

Gema Correa Fernández

Jonathan Fernández Mertanen

Eila Gómez Hidalgo

Elías Méndez García

Alex Enrique Tipán Párraga

1.

CÓMO CALCULAR EL TIEMPO DE EJECUCIÓN

Midiendo
tiempo de
ejecución

CÓMO CALCULAR EL TIEMPO

Usando `high_resolution_clock::now()` obtenemos los tiempos antes y después de lanzar el algoritmo y calculando la diferencia

La variable `tiempo` se usará para calcular la media

```
double tiempo = 0.0f;
for (int i = 0; i < iteraciones; ++i) {

    int *copy = new int[n];
    std::copy(T, T + n, copy);

    assert(copy);

    auto t1 = high_resolution_clock::now();
    algoritmo(copy, n);
    auto t2 = high_resolution_clock::now();

    tiempo += duration_cast<duration<double>>(t2 - t1).count();

    delete[] copy;
}
```

CÓMO CALCULAR EL TIEMPO

Para obtener los datos se ha utilizado este script que genera los archivos con los datos y las gráficas (gnuplot) junto a un archivo Makefile

```
#!/bin/bash

# Argumentos
# $1 nombre del programa a ejecutar
# $2 tamaño de datos inicial
# $3 tamaño de datos de la última iteración
# $4 incremento entre cada iteración
# $5 número de veces que el programa repetirá el cálculo para sacar la media
function generateData() {
    echo "" > $1.dat

    for ((c = $2; c <= $3; c += $4)); do
        ./$1 $c $5 >> $1.dat
    done
}

for I in "burbuja" "heapsort" "insercion" "quicksort" "mergesort" "seleccion"
do
    echo "Calculando los tiempos del algoritmo: $I"
    generateData $I 1000 25000 1000 10
done

echo "Calculando los tiempos del algoritmo: hanoi"
generateData hanoi 10 35 1 10

echo "Calculando los tiempos del algoritmo: floyd"
generateData floyd 100 2500 100 10

for D in "burbuja" "heapsort" "insercion" "quicksort" "mergesort" "seleccion" "hanoi"
"floyd"
do
    echo "Generando gráfico de: $D"
    gnuplot <<< "\
                                set terminal svg; \
                                set output '$D.svg'; \
                                set xlabel 'Tamaño'; \
                                set ylabel 'Tiempo (seg)'; \
                                plot '$D.dat' title 'Eficiencia de $D' with points"
done
```

2.

TABLAS DE TIEMPOS Y GRÁFICAS

¿Cuánto hemos
tardado?

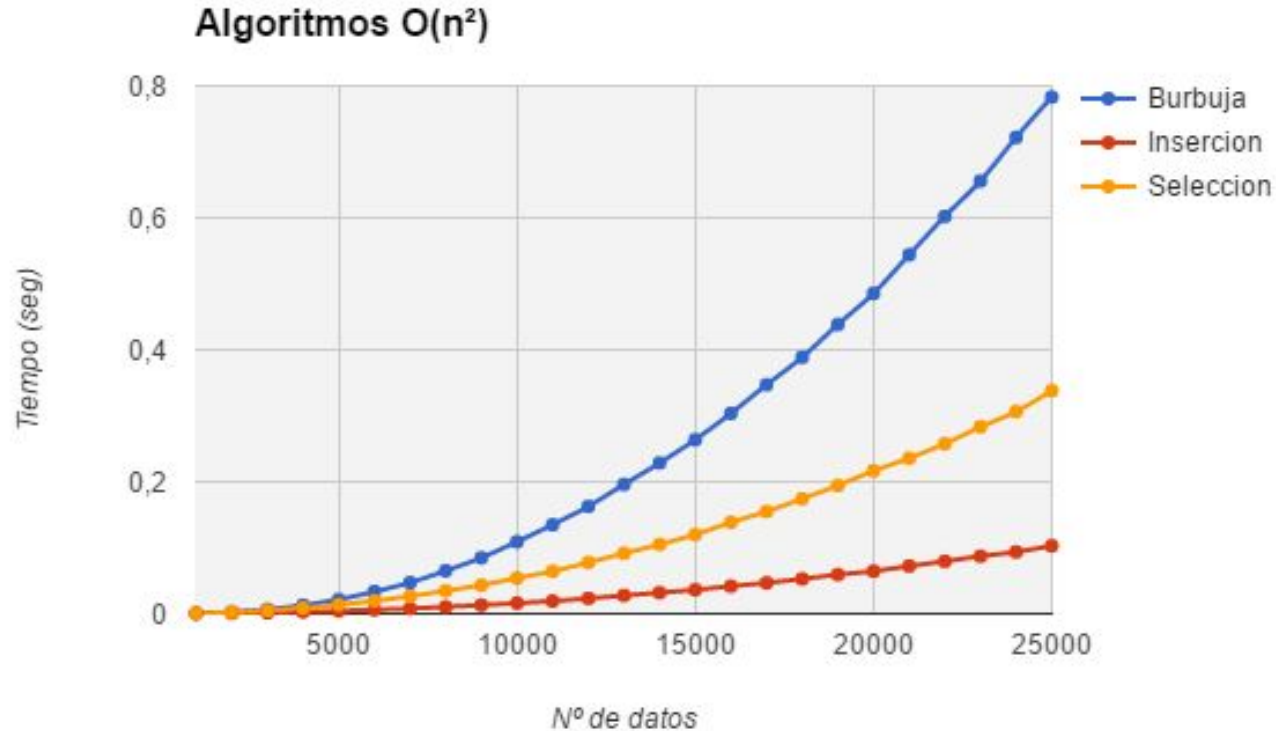
Nº de datos	Burbuja	Inserción	Selección
1000	0,00070484	0,000168415	0,000547321
2000	0,00278126	0,000659599	0,00223252
3000	0,00661067	0,00153272	0,00490875
4000	0,0128572	0,00267263	0,00868611
5000	0,0214832	0,00431276	0,0135702
6000	0,0335224	0,00589975	0,0194873
7000	0,0471374	0,00786774	0,026516
8000	0,06509	0,010325	0,034716
9000	0,0848508	0,0130514	0,0432733
10000	0,109111	0,0160186	0,0541113
11000	0,134892	0,019131	0,0643321
12000	0,162664	0,023165	0,0776892
13000	0,195891	0,0275811	0,0915134
14000	0,228442	0,0318901	0,104928
15000	0,263753	0,0361799	0,119902
16000	0,303609	0,0417343	0,138473
17000	0,347029	0,0466932	0,154819
18000	0,388661	0,0527501	0,174173
19000	0,438423	0,0592765	0,194356
20000	0,485374	0,0644968	0,216541
21000	0,544013	0,072138	0,235675
22000	0,602217	0,0794605	0,257706
23000	0,655833	0,0871451	0,283176
24000	0,721958	0,0936699	0,306367
25000	0,783014	0,102941	0,338052

TABLA



$O(n^2)$

GRÁFICA ALGORITMOS $O(n^2)$



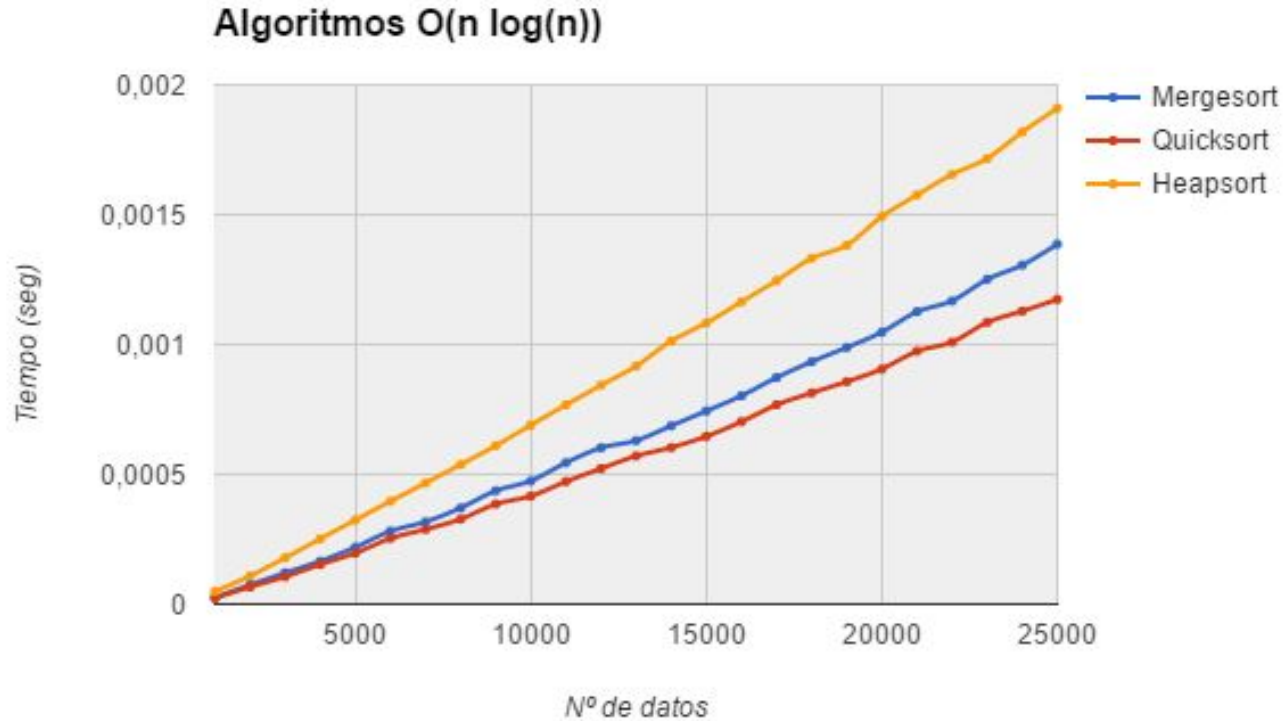
Nº de datos	Mergesort	Quicksort	Heapsort
1000	0,0000254	2,32E-05	4,84E-05
2000	0,0000743	6,61E-05	0,000108417
3000	0,0001207	0,000105517	0,000178247
4000	0,0001651	0,000152682	0,000251412
5000	0,0002191	0,00019645	0,000323969
6000	0,0002814	0,000253365	0,000396304
7000	0,0003167	0,000287885	0,000466316
8000	0,0003703	0,000327304	0,000537423
9000	0,0004369	0,000385544	0,000610291
10000	0,0004746	0,000415927	0,000689575
11000	0,0005446	0,000471943	0,000767739
12000	0,0006024	0,000522267	0,000843046
13000	0,0006297	0,000570751	0,000916754
14000	0,0006869	0,000603042	0,00101354
15000	0,0007438	0,000646043	0,00108273
16000	0,0008024	0,000702551	0,00116413
17000	0,0008726	0,000767774	0,00124579
18000	0,0009337	0,000813343	0,00133197
19000	0,0009884	0,000856854	0,00138204
20000	0,0010473	0,00090537	0,00149273
21000	0,0011262	0,000974075	0,00157585
22000	0,0011676	0,00100891	0,00165497
23000	0,0012511	0,00108535	0,00171587
24000	0,0013058	0,00112833	0,0018183
25000	0,0013863	0,00117366	0,00190969

TABLA



$O(n \log n)$

GRÁFICA ALGORITMOS $O(n \log(n))$



TABLA



Nº de datos	Floyd
100	0,000927308
200	0,00686802
300	0,0223873
400	0,0519861
500	0,101854
600	0,176301
700	0,277682
800	0,41186
900	0,5953
1000	0,827693
1100	1,159
1200	1,54931
1300	2,02421
1400	2,64131
1500	3,2443
1600	3,97622
1700	4,79084
1800	5,67815
1900	6,67535
2000	7,77094
2100	9,03474
2200	10,1211
2300	11,6046
2400	13,2707
2500	15,2813

$O(n^3)$

GRÁFICA ALGORITMOS $O(n^3)$



Nº de datos	Hanoi
10	1,50E-06
11	2,72E-06
12	5,19E-06
13	1,02E-05
14	1,94E-05
15	3,84E-05
16	8,33E-05
17	0,000162479
18	0,000332946
19	0,00067568
20	0,00138661
21	0,00273687
22	0,00553412
23	0,0110794
24	0,0220678
25	0,0438211
26	0,0867785
27	0,171589
28	0,340834
29	0,67166
30	1,33313
31	2,64708
32	5,30209
33	10,6107
34	21,4292
35	42,8149

TABLA



$O(2^n)$

GRÁFICA ALGORITMOS $O(2^n)$



3.

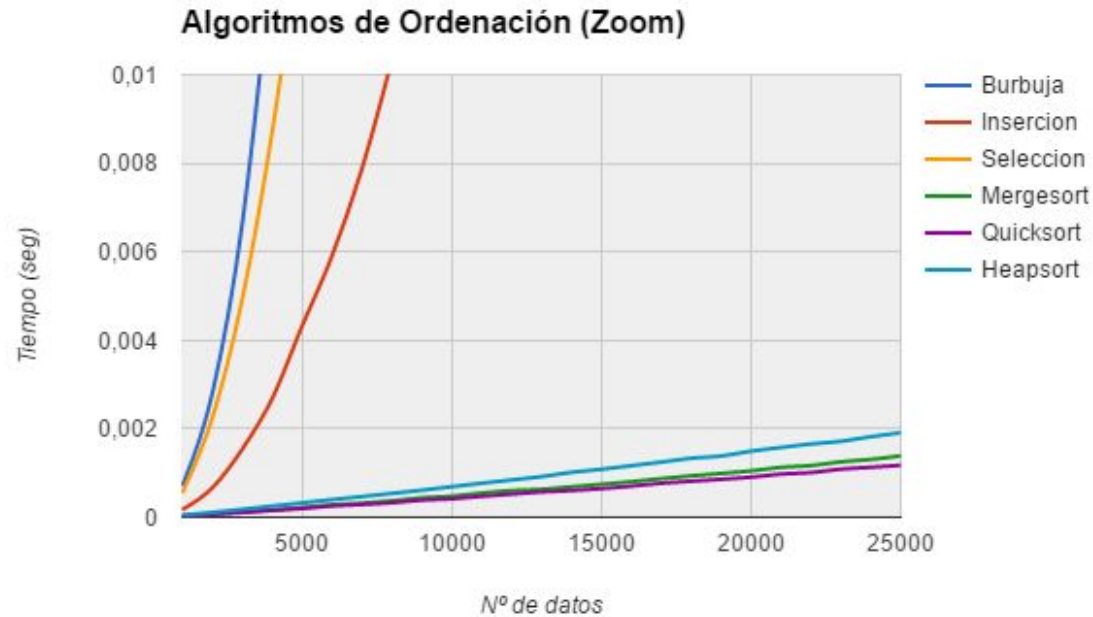
GRÁFICAS DE COMPARACIÓN

Vamos a
compararlos
todos

GRÁFICA ALGORITMOS DE ORDENACIÓN



GRÁFICA CON ZOOM





¡GRACIAS A TODOS!

¿Preguntas?