

Algorítmica: práctica 1

Análisis de la eficiencia de algoritmos

Sofía Almeida Bruno

Antonio Coín Castro

María Victoria Granados Pozo

Miguel Lentisco Ballesteros

José María Martín Luque

16 de marzo de 2017

Objetivo

Estudiar tanto la eficiencia teórica como la eficiencia empírica de 8 algoritmos.

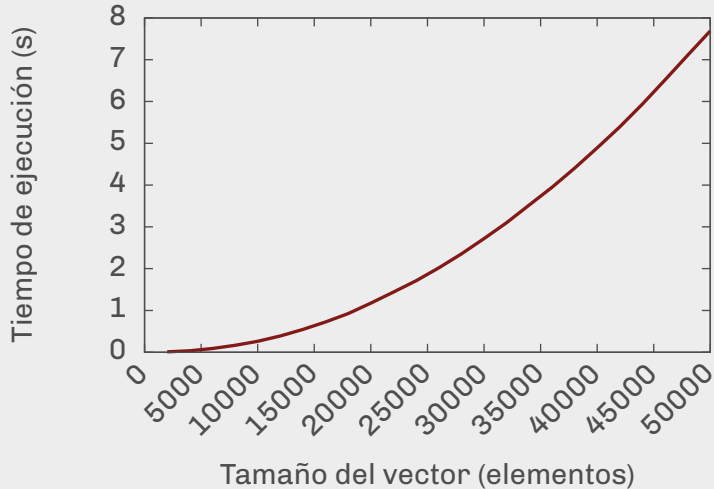
- Burbuja, Inserción y Selección
- Mergesort, Quicksort y Heapsort
- Floyd y Hanoi

Burbuja

Revisa cada elemento de la lista con el siguiente, intercambiándose de posición si no están en el orden correcto.

Es $O(n^2)$.

Eficiencia empírica burbuja-linux-00

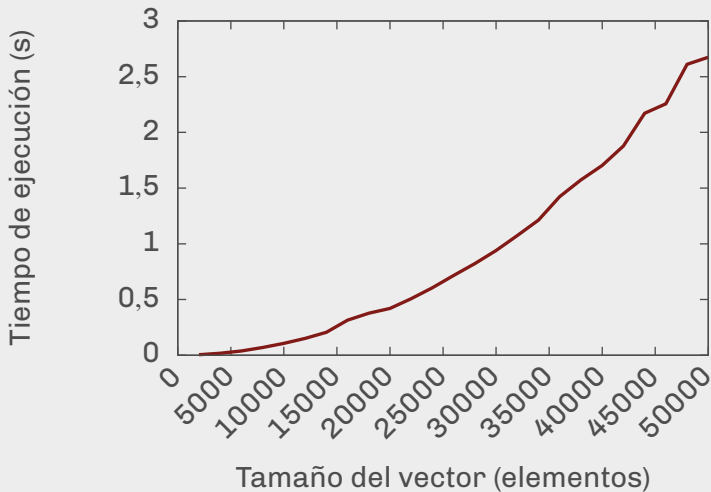


Inserción

Consideramos el elemento N-ésimo de la lista y lo ordenamos respecto de los elementos desde el primero hasta el N-1-ésimo.

Es $O(n^2)$

Eficiencia empírica insercion-linux-00

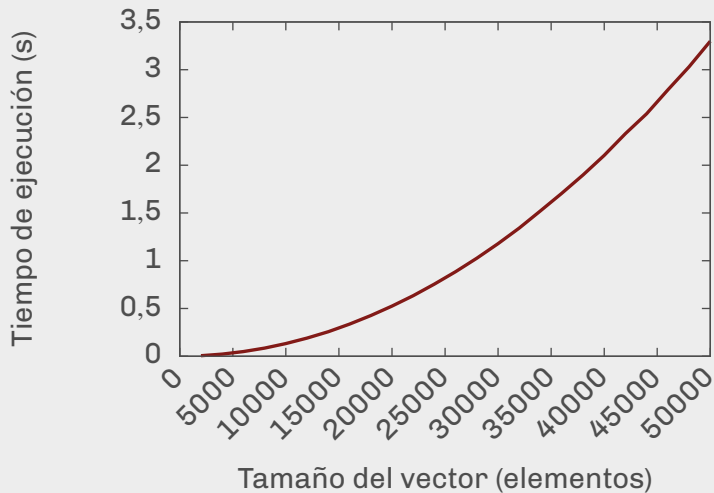


Selección

Consiste en encontrar el menor de todos los elementos de la lista e intercambiarlo con el de la primera posición. Luego con el segundo, y así sucesivamente hasta ordenarlo todo.

Es $O(n^2)$

Eficiencia empírica seleccion-linux-00



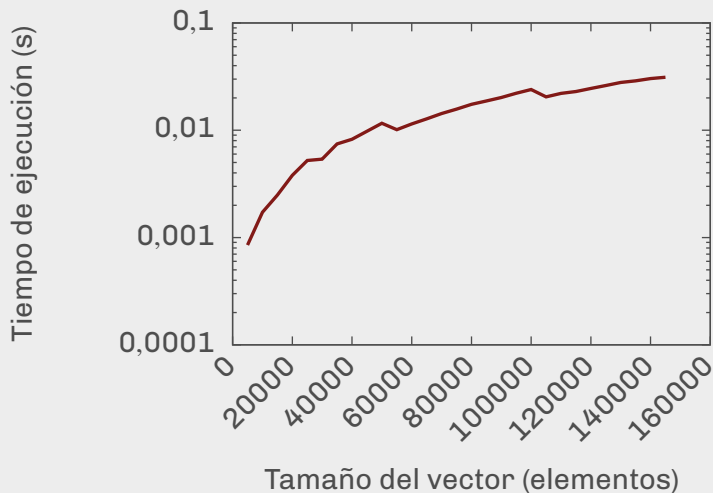
Mergesort

Se basa en la técnica de *divide y vencerás*.

- Se divide la lista a ordenar en dos sublistas de la mitad de tamaño.
- Se ordena cada sublista de forma recursiva.
- Si el tamaño de una sublista es 0 o 1 entonces ya está ordenada.
- Se unen todas las sublistas en una sola.

Es $O(n \log n)$

Eficiencia empírica mergesort-linux-00



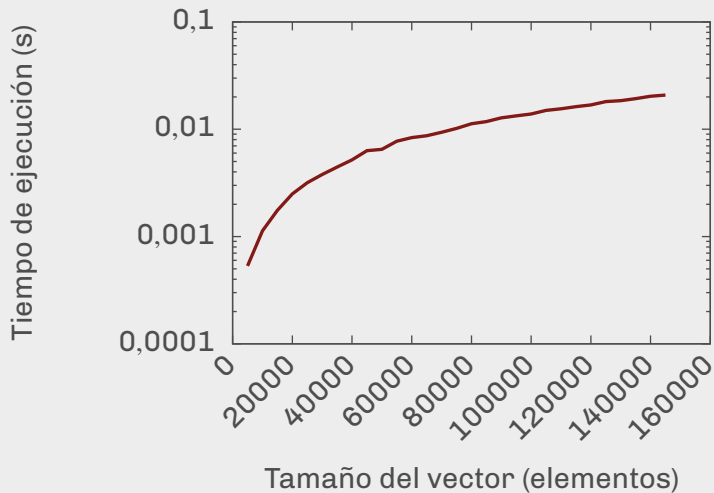
Quicksort

Se basa en la técnica de *divide y vencerás*.

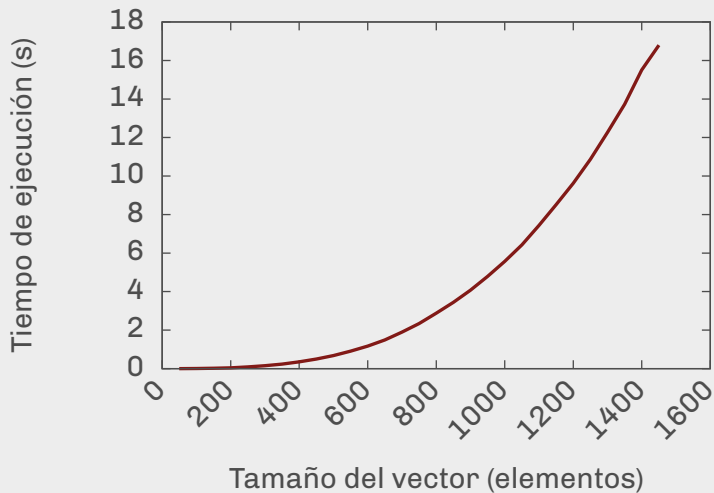
- Elegimos un elemento de la lista, el *pivote*.
- Se ordena la lista, dejando los elementos mayores a la derecha del pivote y los menores a la izquierda.
- Realizamos el proceso recursivamente en las dos sublistas que nos quedan (derecha e izquierda) hasta que tengan 0 o 1 elemento.

Es $O(n \log n)$

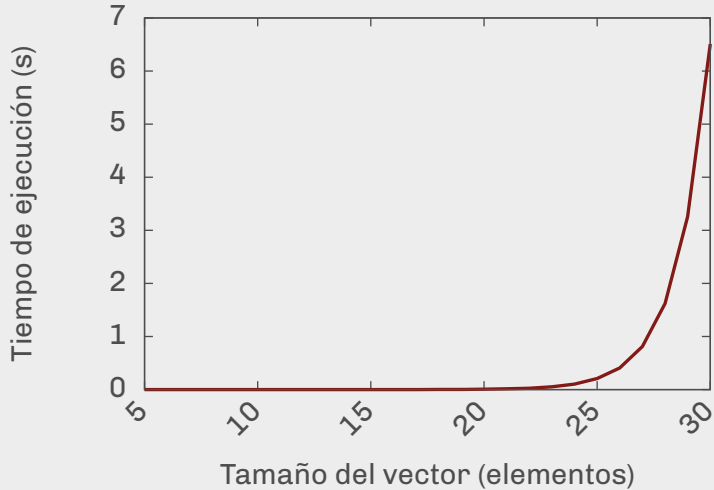
Eficiencia empírica quicksort-linux-00



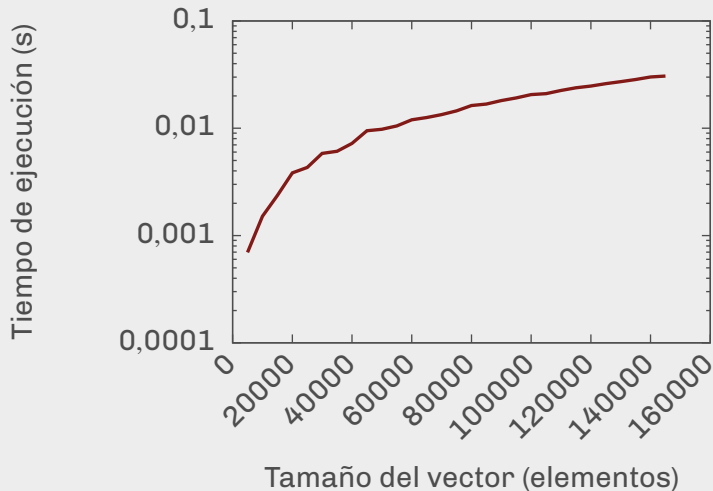
Eficiencia empírica floyd-linux-00



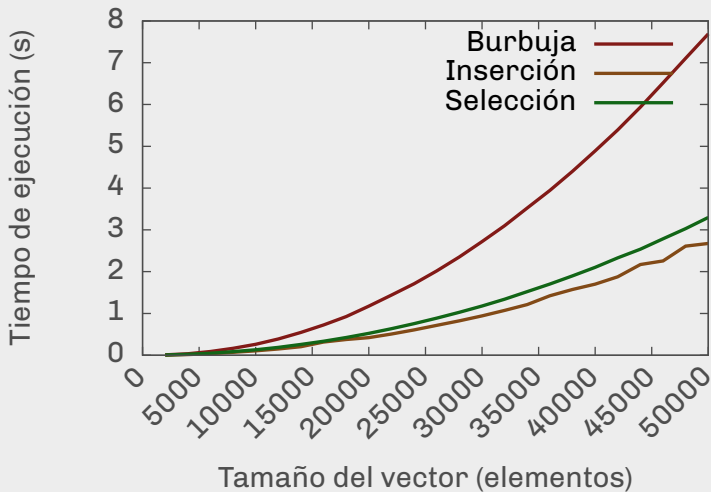
Eficiencia empírica hanoi-linux-00



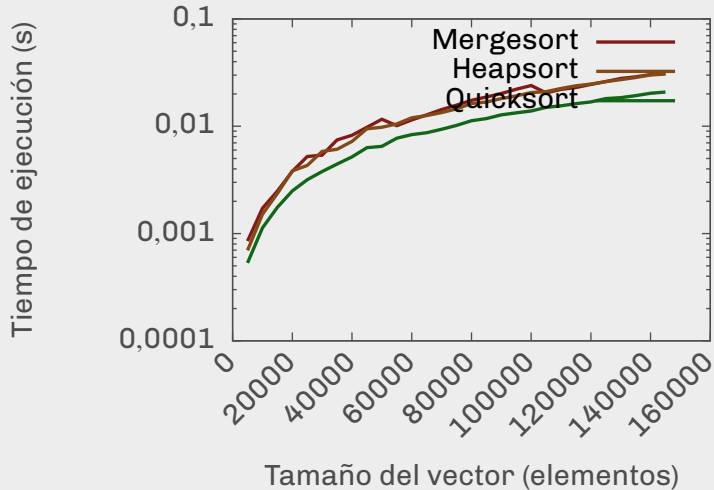
Eficiencia empírica heapsort-linux-00



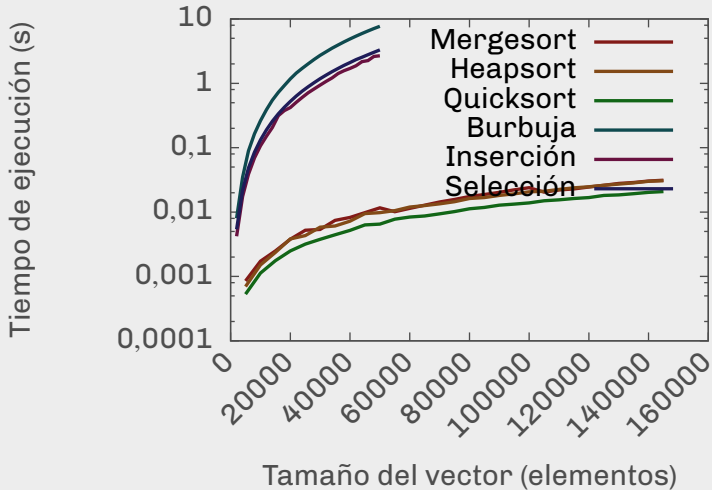
Algoritmos de ordenación $O(n^2)$



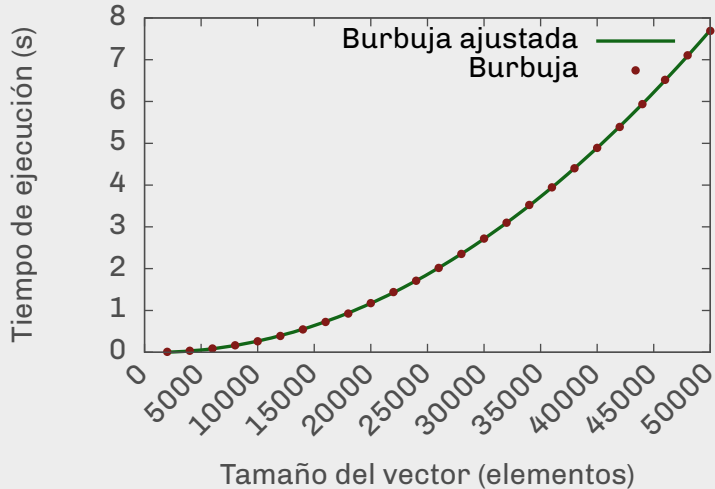
Algoritmos de ordenación $O(n \log n)$



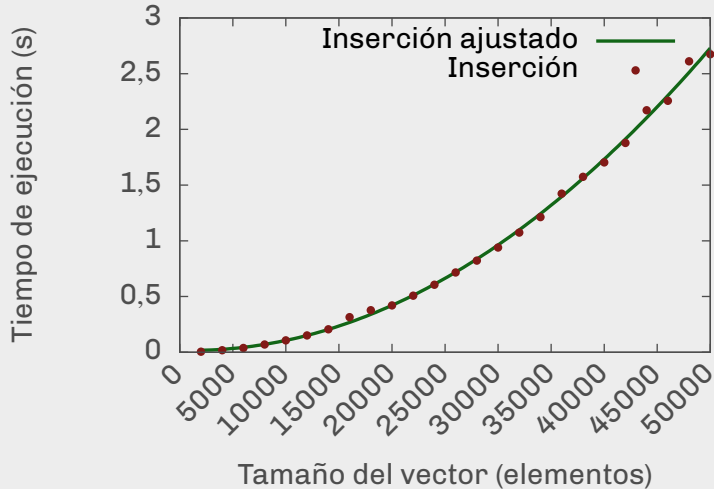
Algoritmos de ordenación



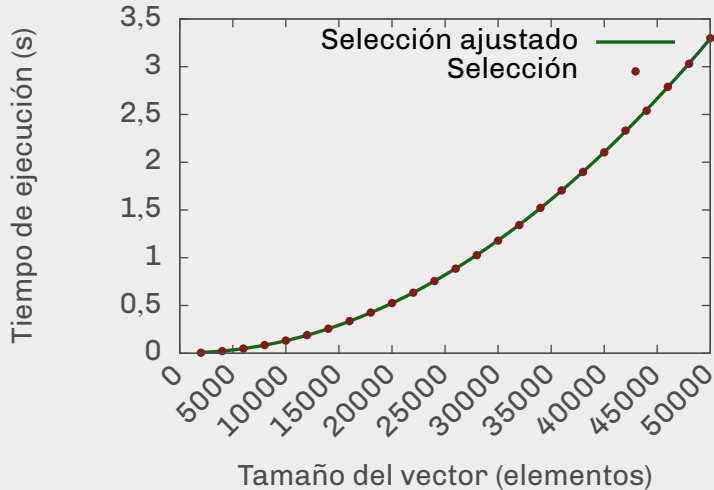
Ajuste Burbuja

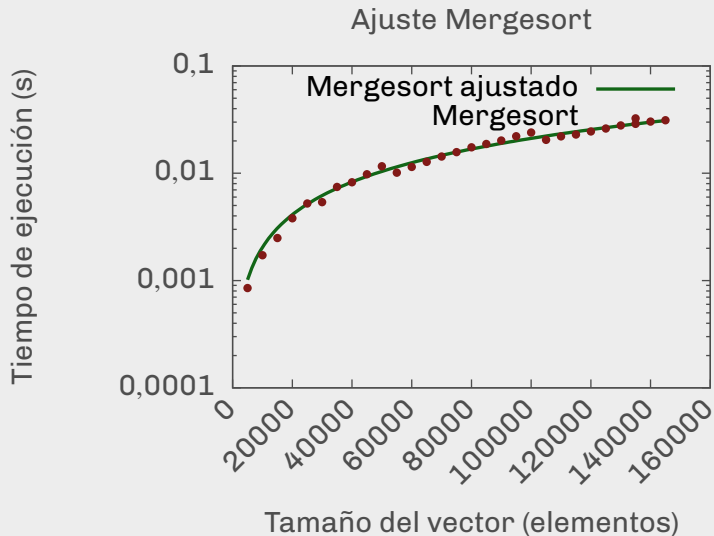


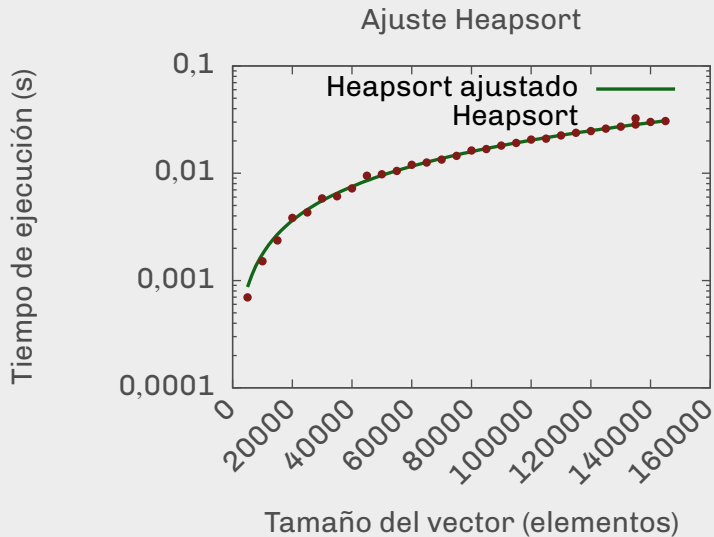
Ajuste Inserción

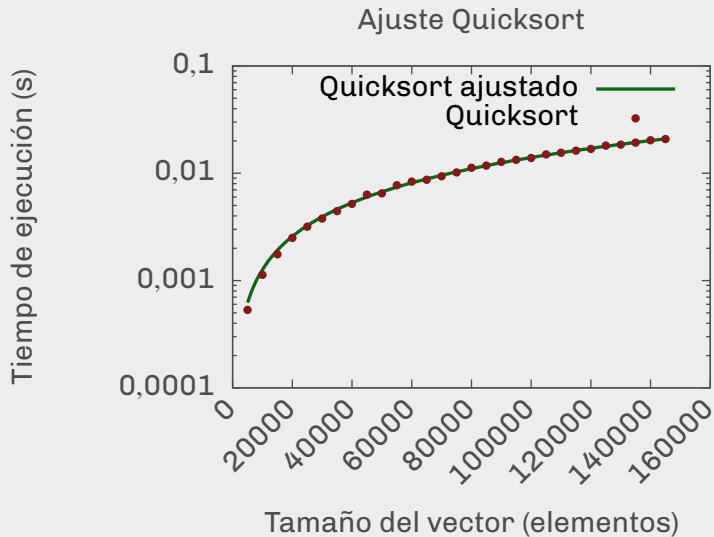


Ajuste Selección

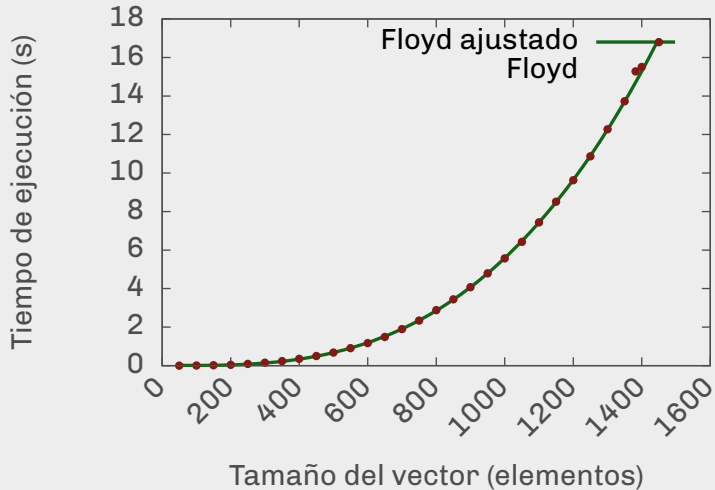




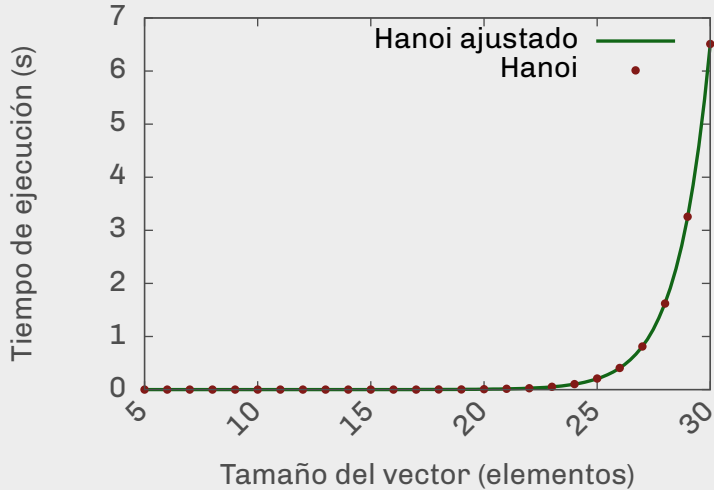




Ajuste Floyd



Ajuste Hanoi



Elementos	Burbuja	Selección	Inserción
2,000	$8,02 \cdot 10^{-3}$	$5,4 \cdot 10^{-3}$	$4,21 \cdot 10^{-3}$
4,000	$3,5 \cdot 10^{-2}$	$2,17 \cdot 10^{-2}$	$1,74 \cdot 10^{-2}$
6,000	$8,93 \cdot 10^{-2}$	$4,84 \cdot 10^{-2}$	$3,87 \cdot 10^{-2}$
8,000	0,16	$8,52 \cdot 10^{-2}$	$6,94 \cdot 10^{-2}$
10,000	0,26	0,13	0,11
12,000	0,39	0,19	0,15
14,000	0,55	0,26	0,21
16,000	0,73	0,34	0,32
18,000	0,93	0,43	0,38
20,000	1,18	0,52	0,42
22,000	1,44	0,63	0,51
24,000	1,71	0,76	0,61
26,000	2,02	0,89	0,72
28,000	2,35	1,03	0,82
30,000	2,72	1,18	0,94
32,000	3,1	1,34	1,07
34,000	3,53	1,52	1,21
36,000	3,95	1,71	1,42
38,000	4,4	1,9	1,57
40,000	4,89	2,1	1,7
42,000	5,39	2,33	1,88
44,000	5,94	2,54	2,17
46,000	6,52	2,79	2,26
48,000	7,11	3,03	2,61
50,000	7,69	3,3	2,67

Algoritmos que
son $O(n^2)$