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

15 de marzo de 2017

Test

Hola a todos
Me gustan los ponies
Helicóptero

$$f_x = x^4 + 5$$

- Unicornio
- Pony
- Caballo

1. Unicornio
2. Pony
3. Caballo

Teorema

Esto es un teorema.

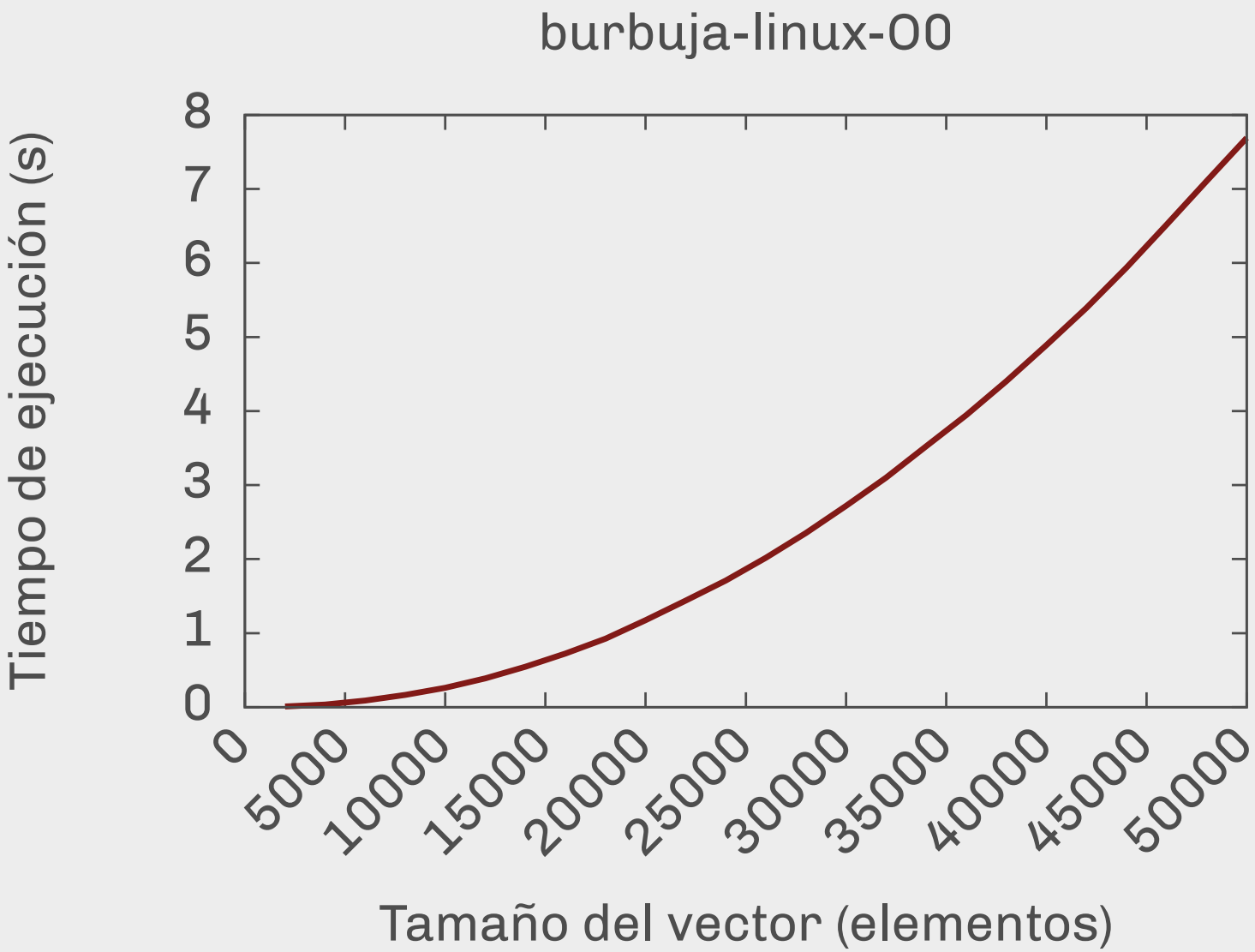
Corolario

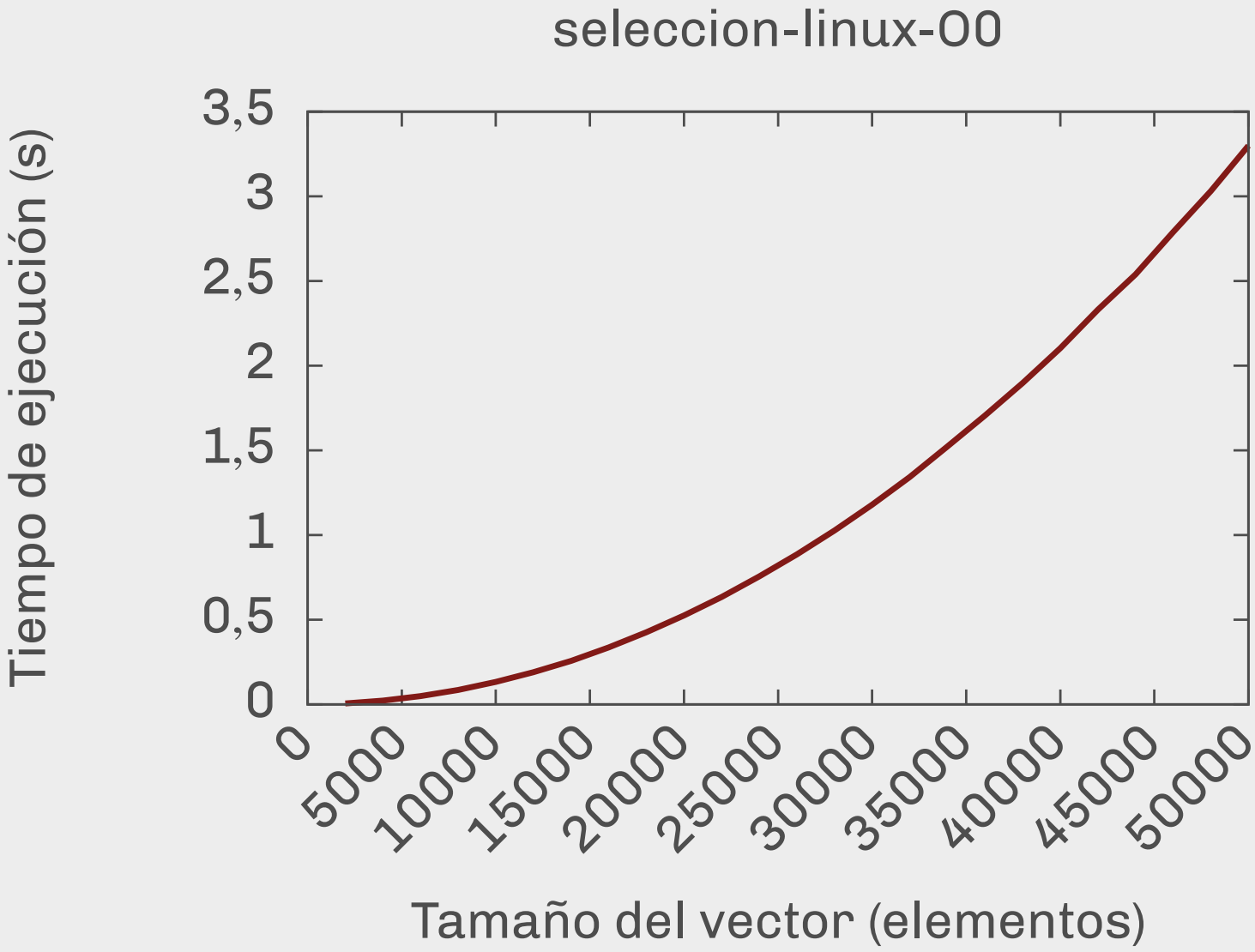
Esto es un corolario.

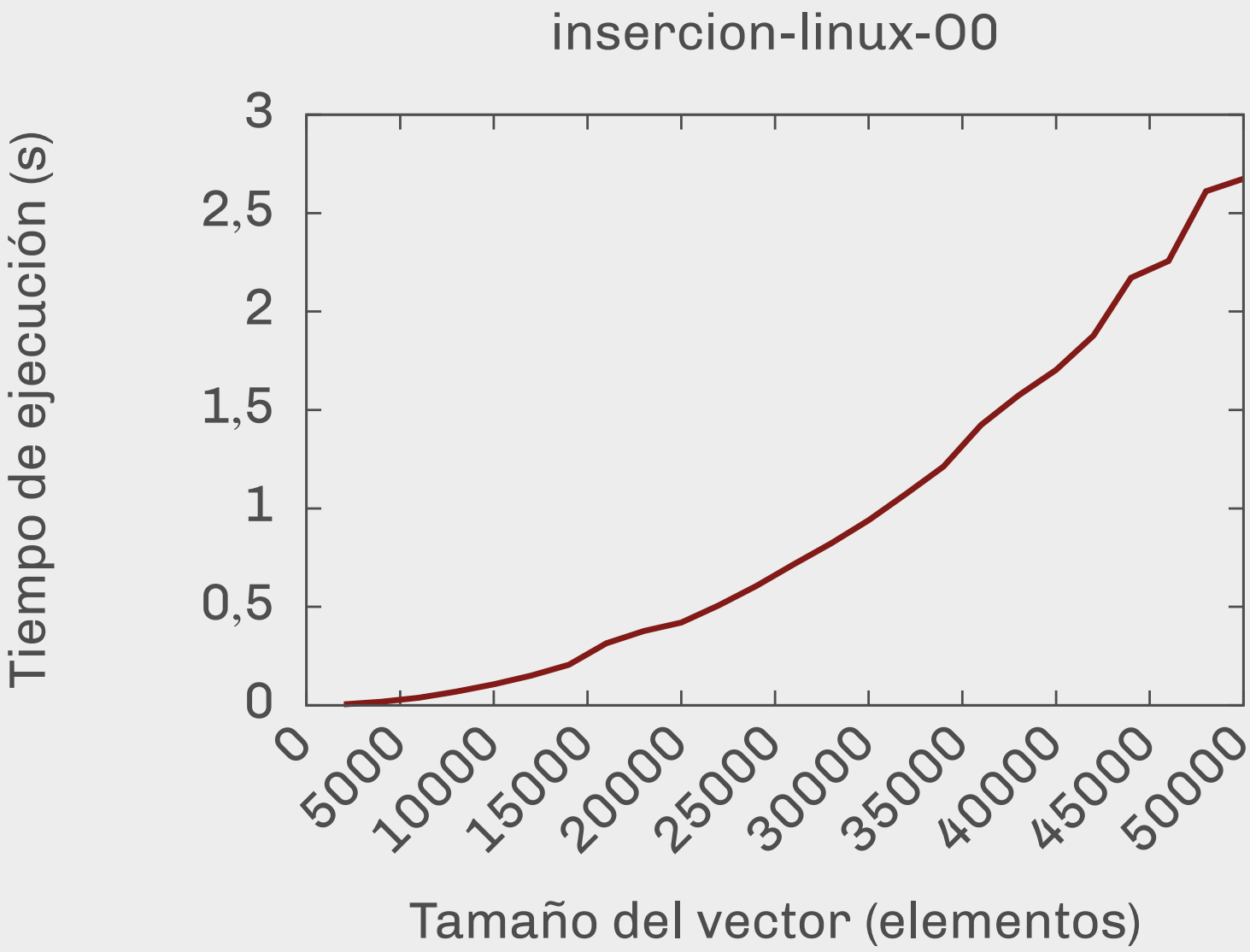
Demostración.

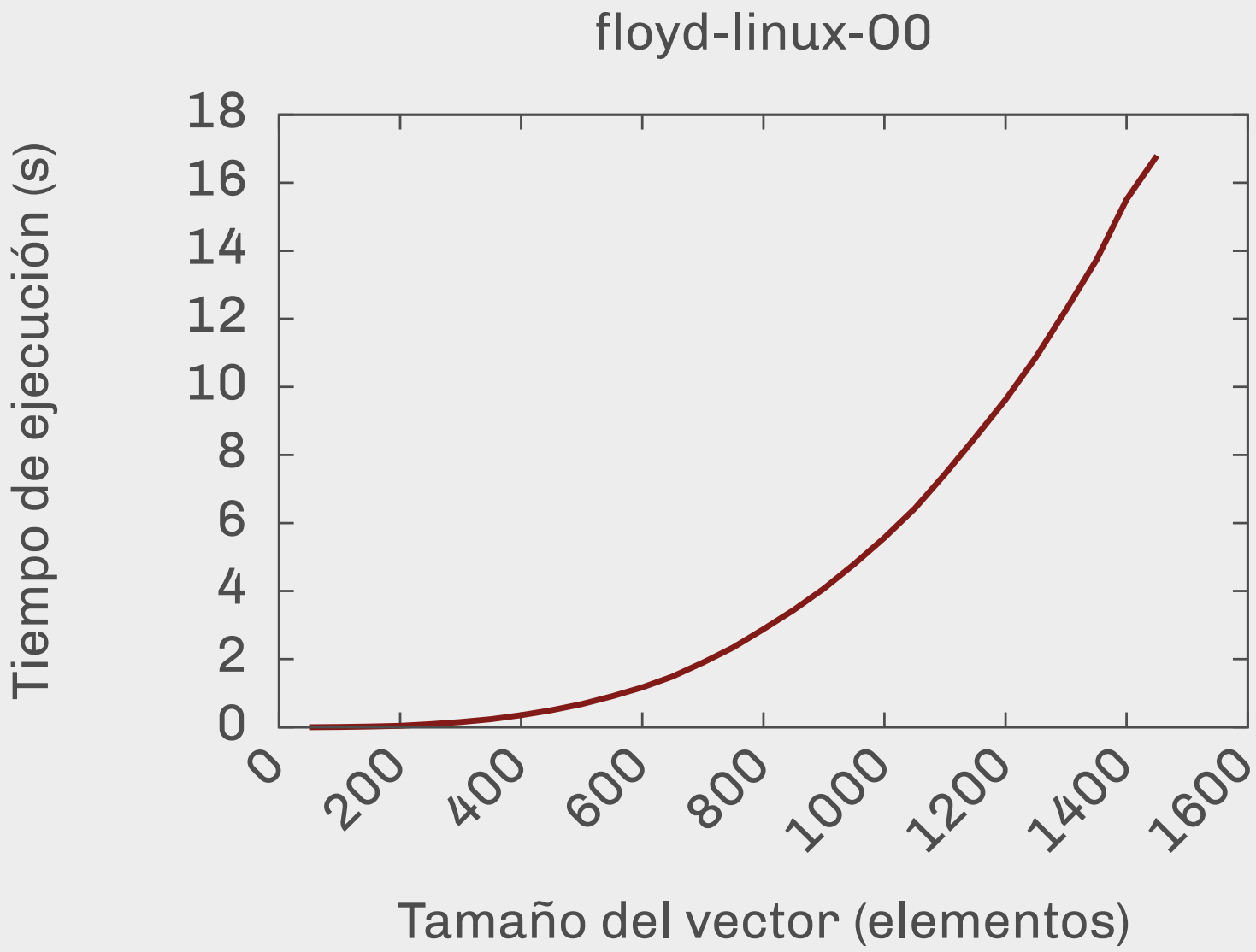
$$d((t,x),(t_0,x_0)) = \sqrt{(t-t_0)^2 + (x-x_0)^2} < \varepsilon_0$$

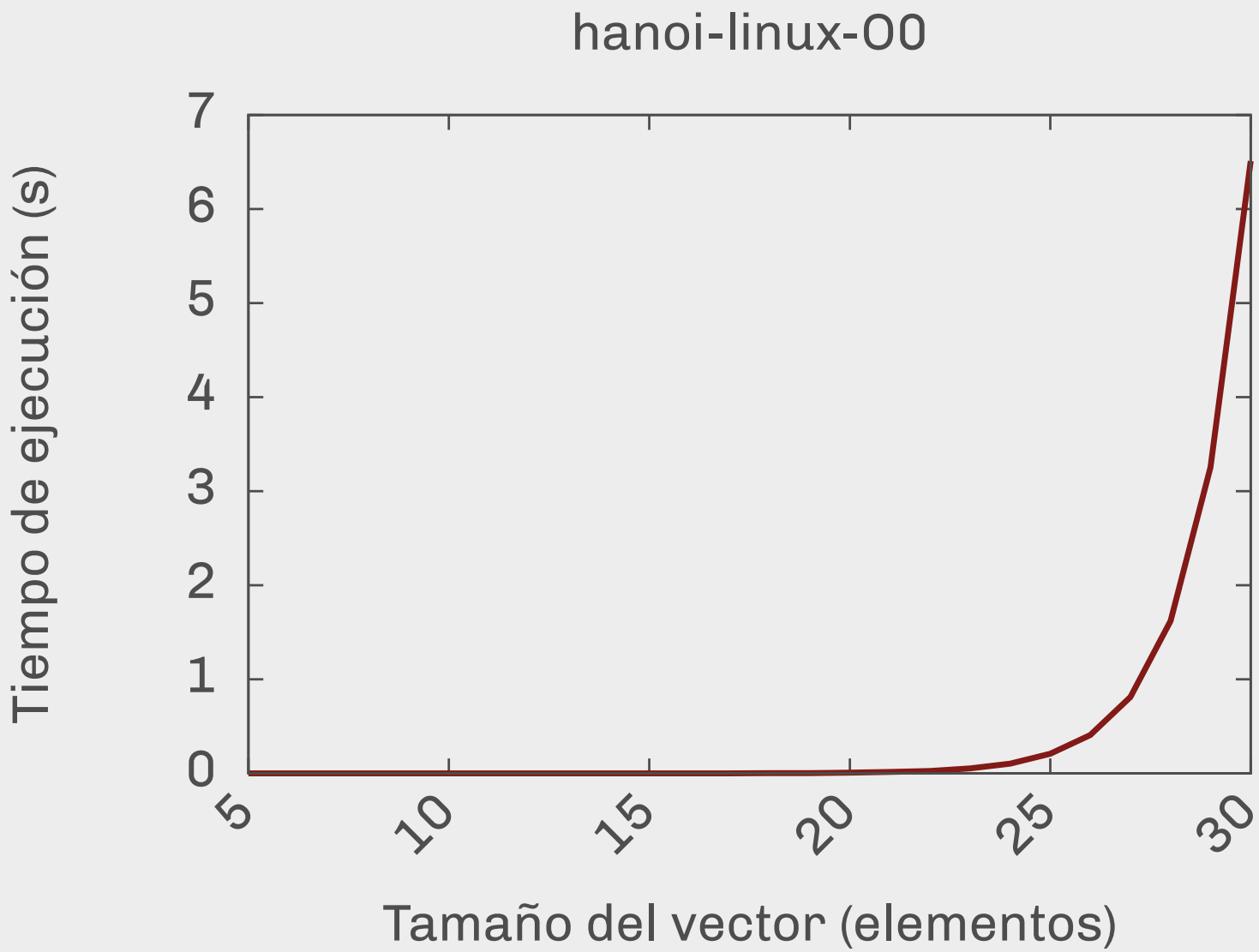
□

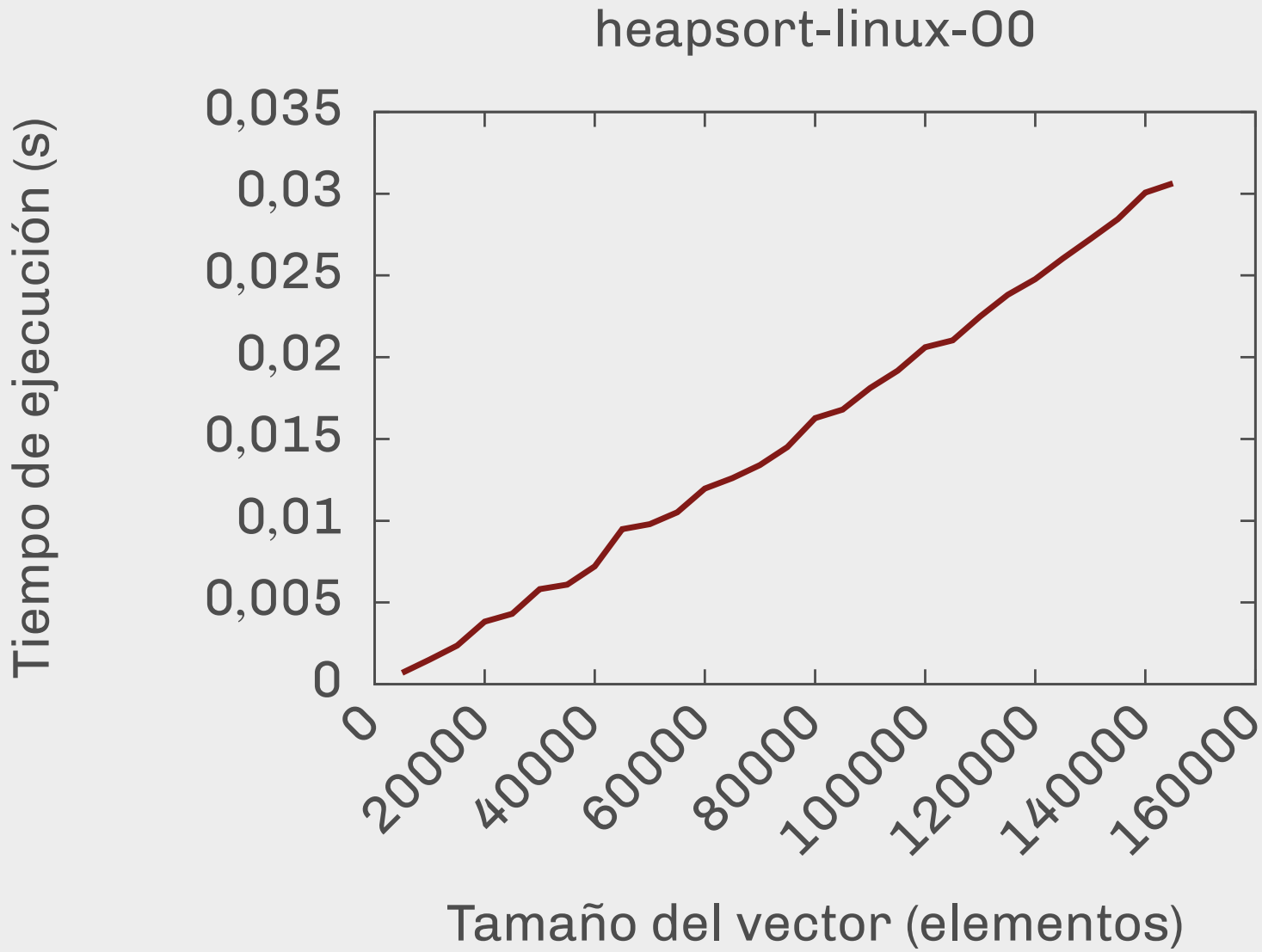

burbuja-linux-O0
Tiempo de ejecución (s)
0
1
2
3
4
5
6
7
8
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000
Tamaño del vector (elementos)

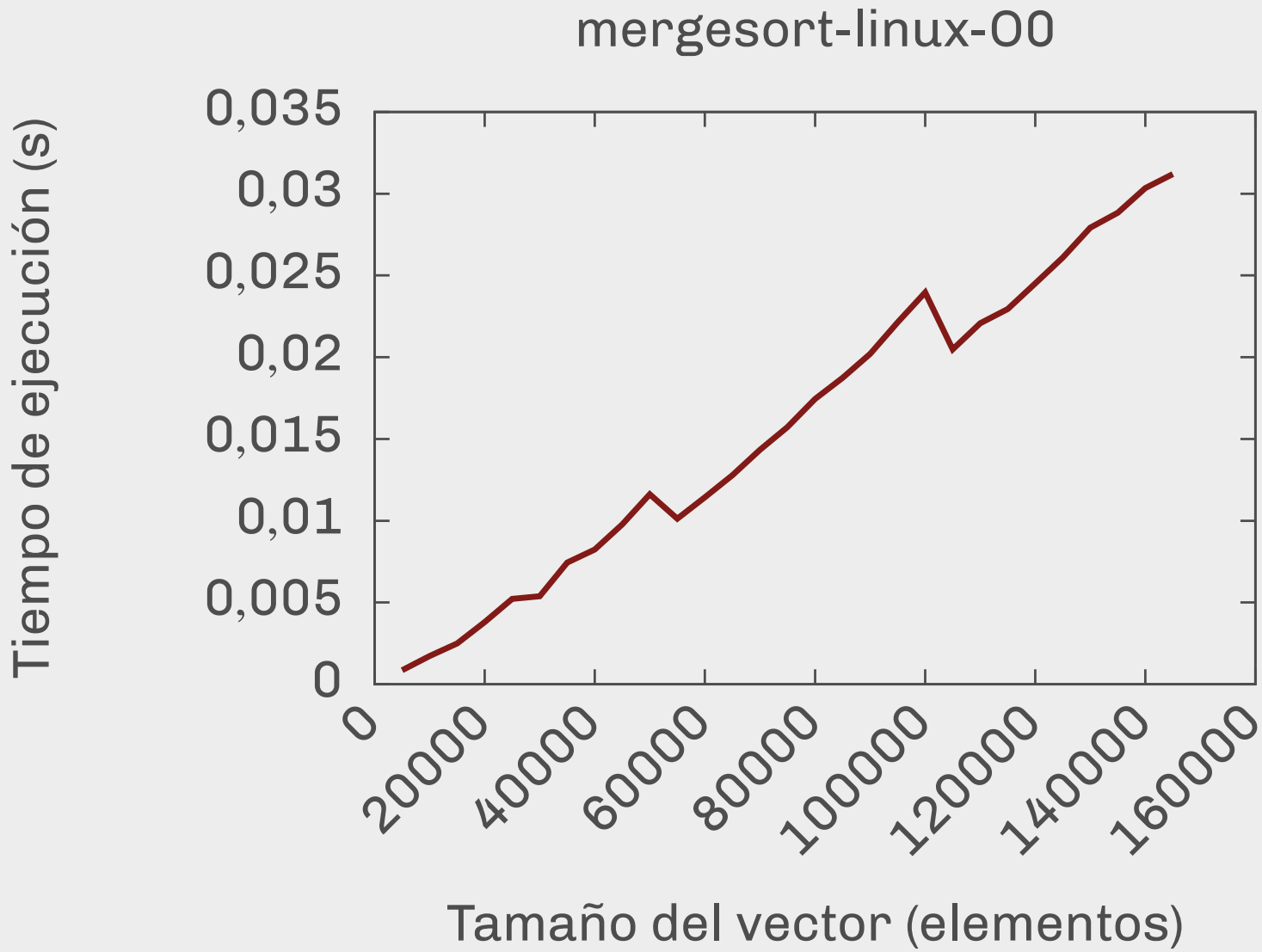

seleccion-linux-O0
Tiempo de ejecución (s)
3,5
3
2,5
2
1,5
1
0,5
0
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000
Tamaño del vector (elementos)

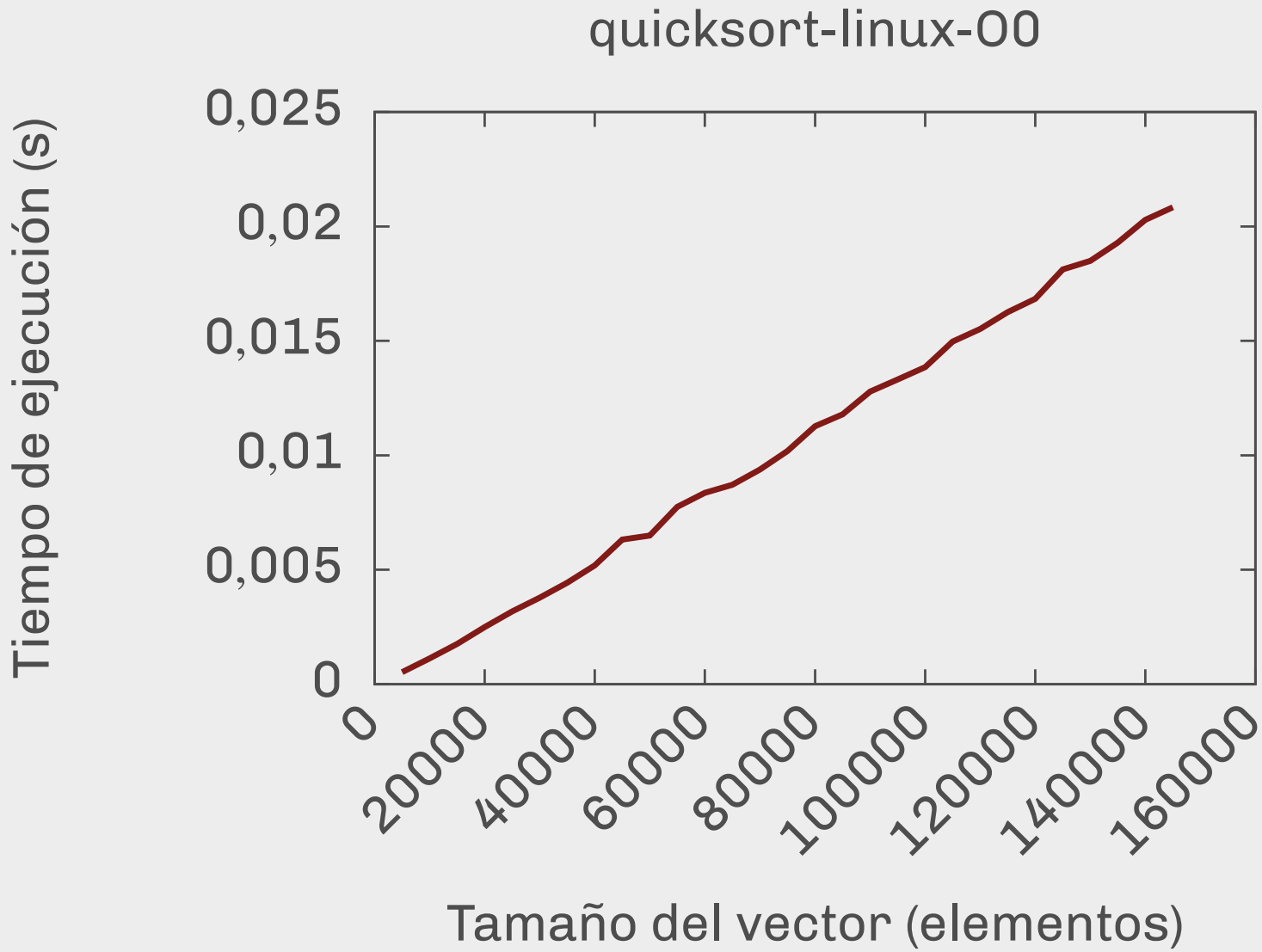

insercion-linux-O0
Tiempo de ejecución (s)
3
2,5
2
1,5
1
0,5
0
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000
Tamaño del vector (elementos)

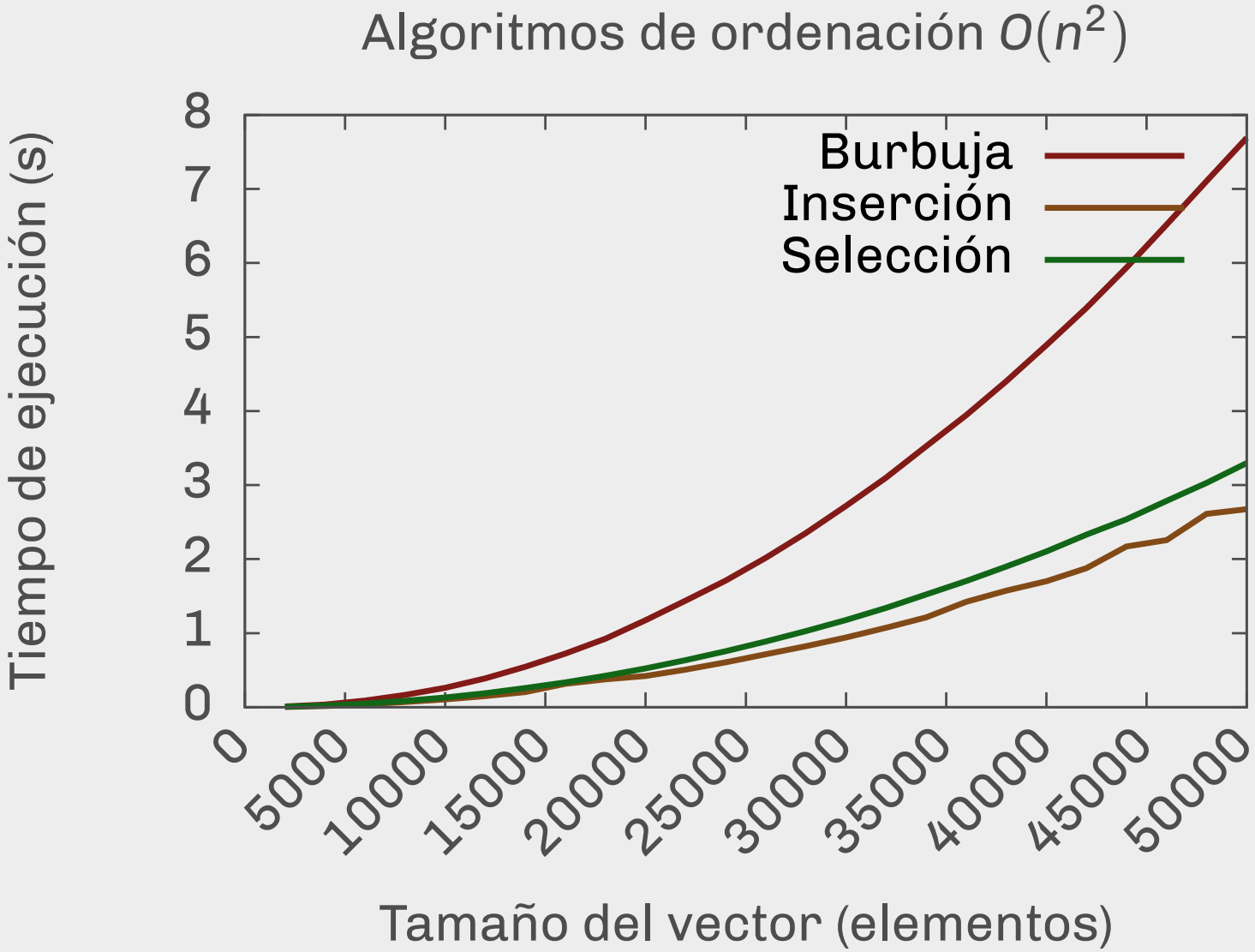

floyd-linux-O0
Tiempo de ejecución (s)
0
2
4
6
8
10
12
14
16
18
0
200
400
600
800
1000
1200
1400
1600
Tamaño del vector (elementos)

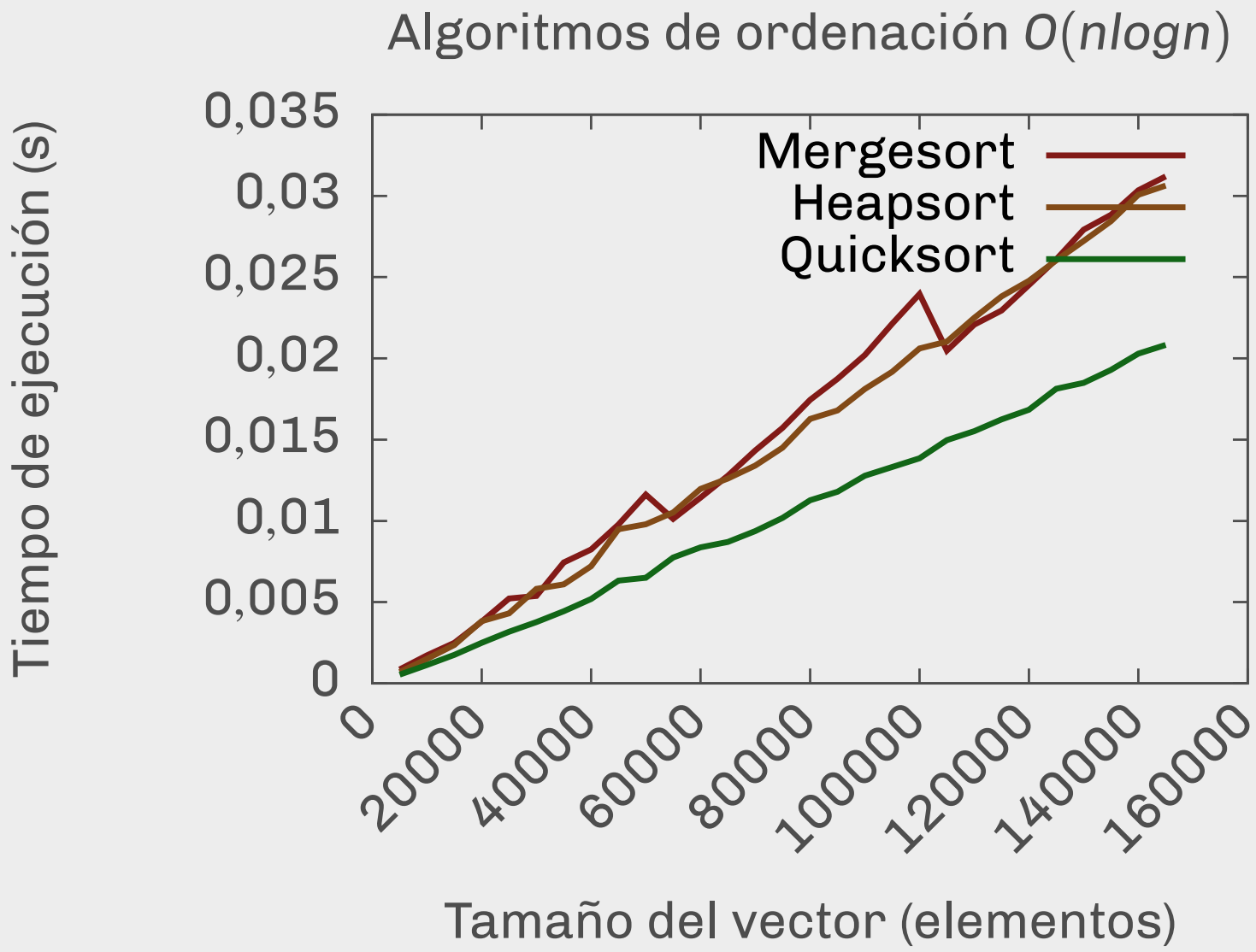

hanoi-linux-O0
Tiempo de ejecución (s)
0
1
2
3
4
5
6
7
5
10
15
20
25
30
Tamaño del vector (elementos)

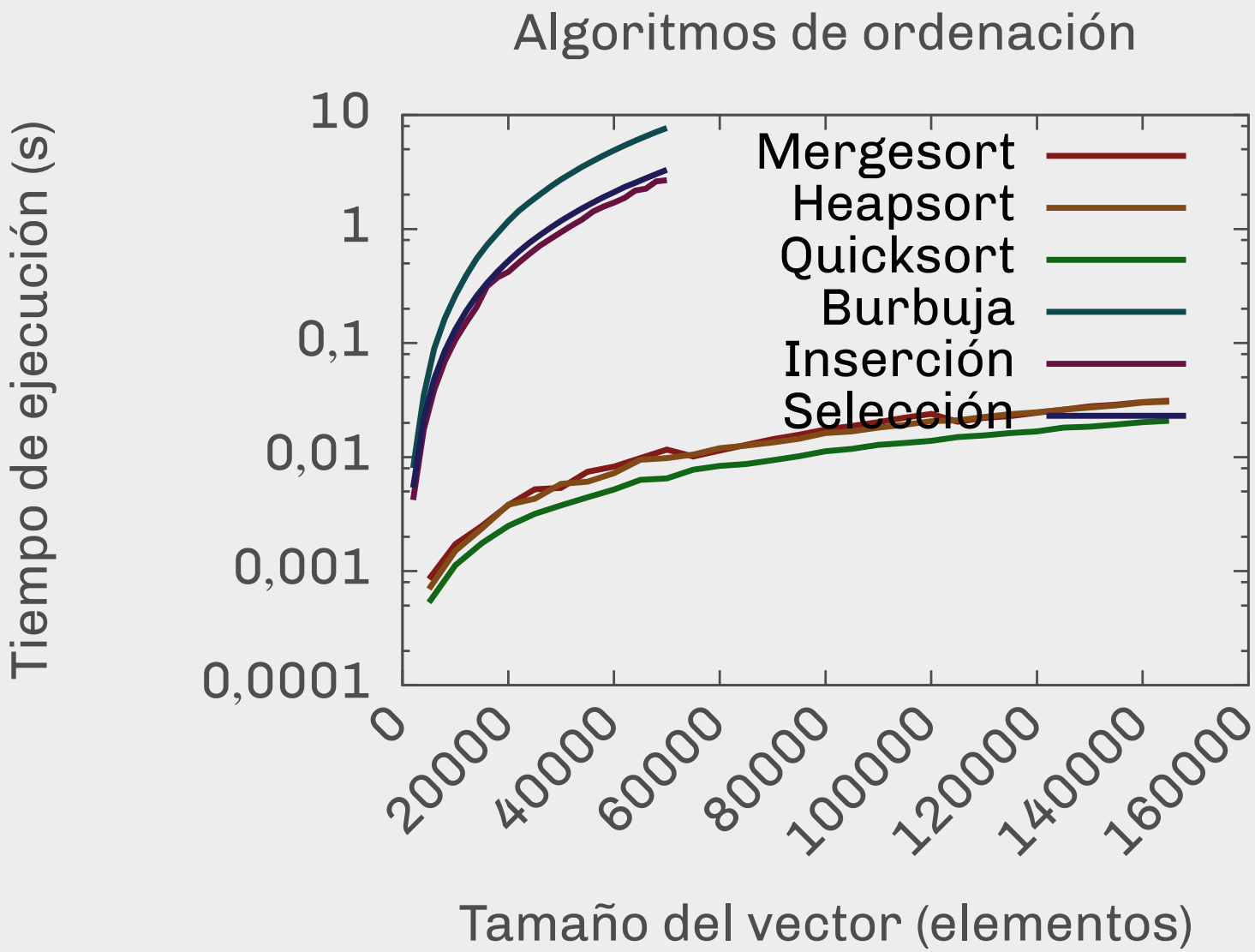

heapsort-linux-O0
Tiempo de ejecución (s)
0,035
0,03
0,025
0,02
0,015
0,01
0,005
0
0
20000
40000
60000
80000
100000
120000
140000
160000
Tamaño del vector (elementos)

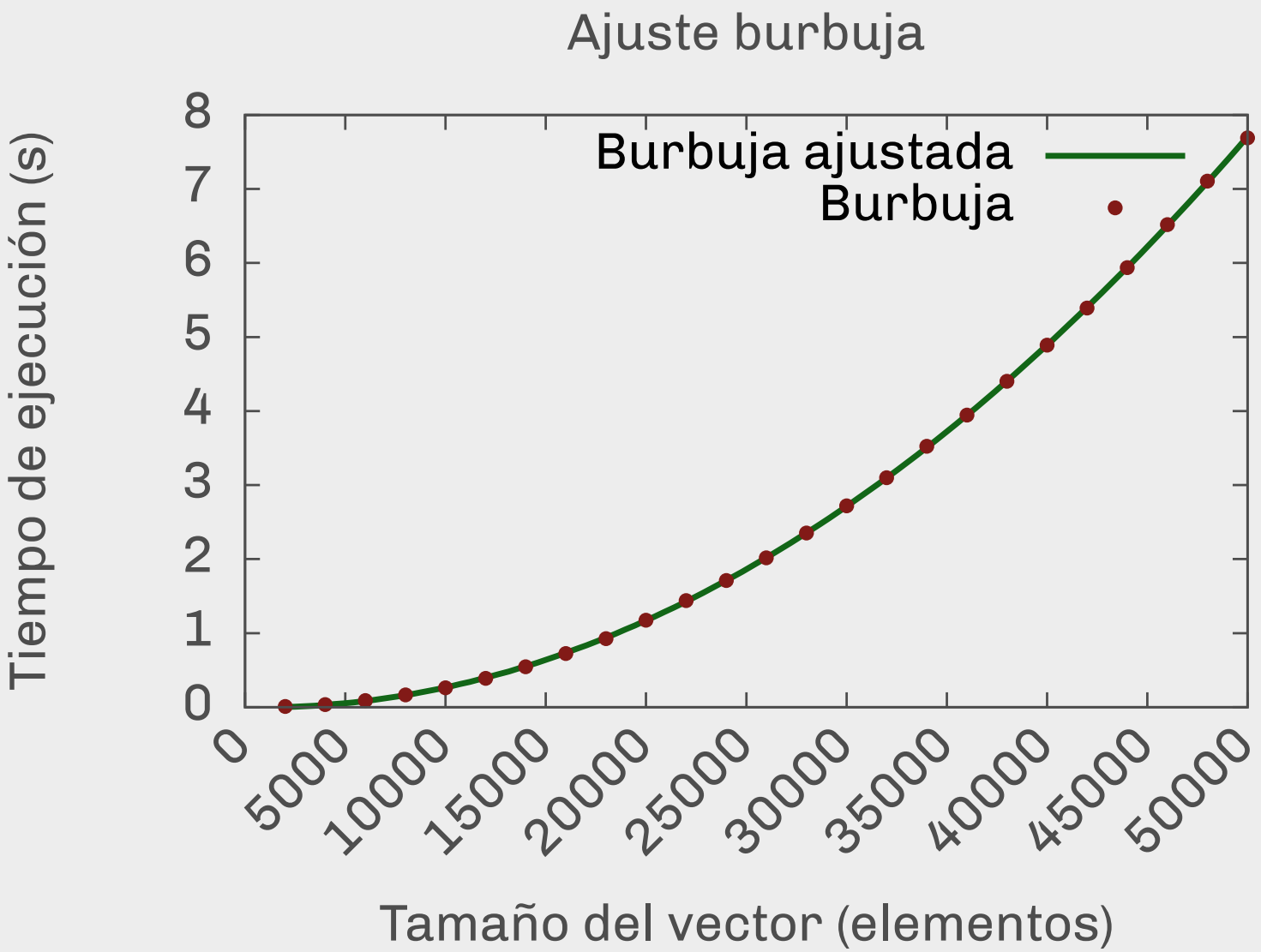

mergesort-linux-O0
Tiempo de ejecución (s)
0,035
0,03
0,025
0,02
0,015
0,01
0,005
0
0
20000
40000
60000
80000
100000
120000
140000
160000
Tamaño del vector (elementos)

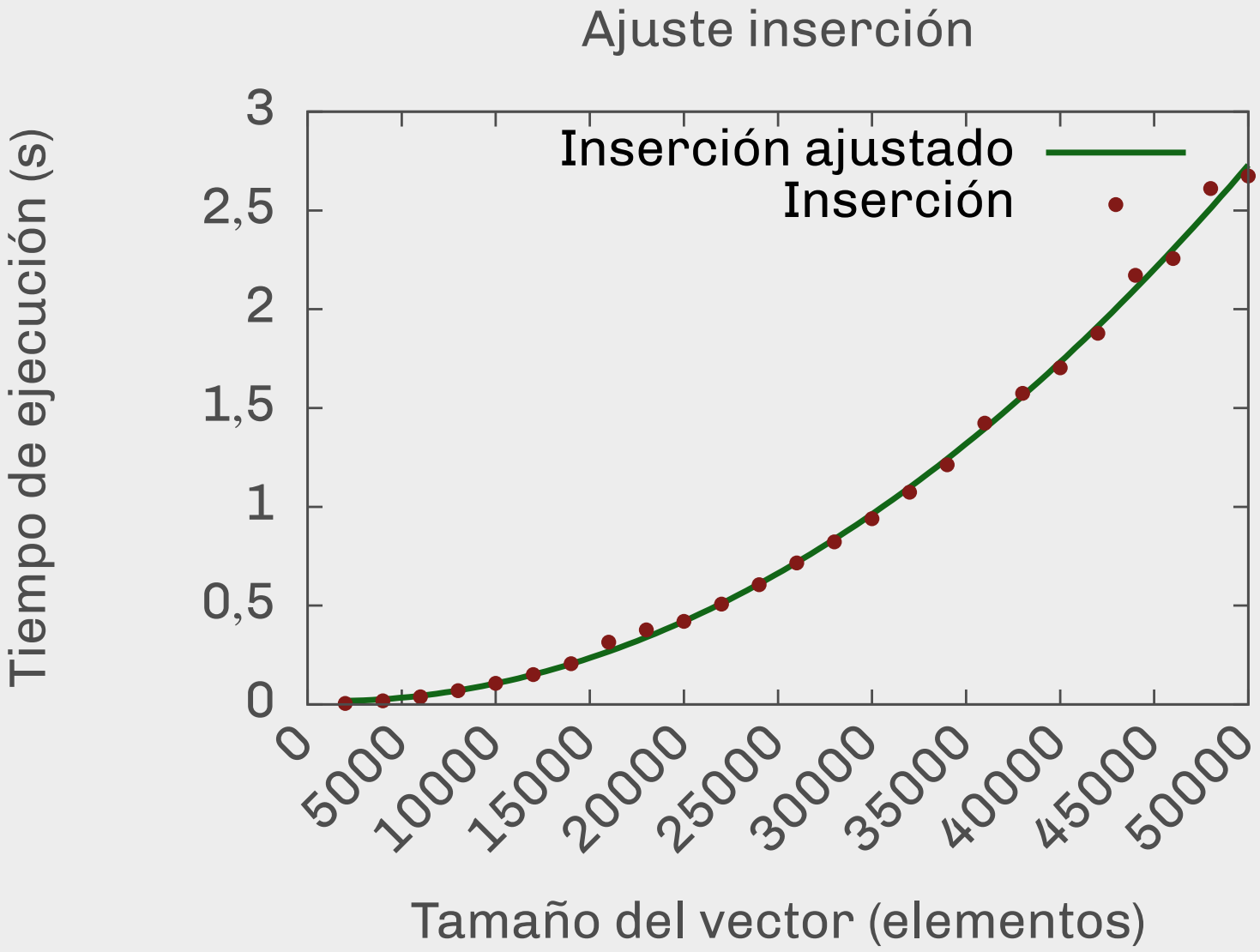

quicksort-linux-O0
Tiempo de ejecución (s)
0,025
0,02
0,015
0,01
0,005
0
0
20000
40000
60000
80000
100000
120000
140000
160000
Tamaño del vector (elementos)

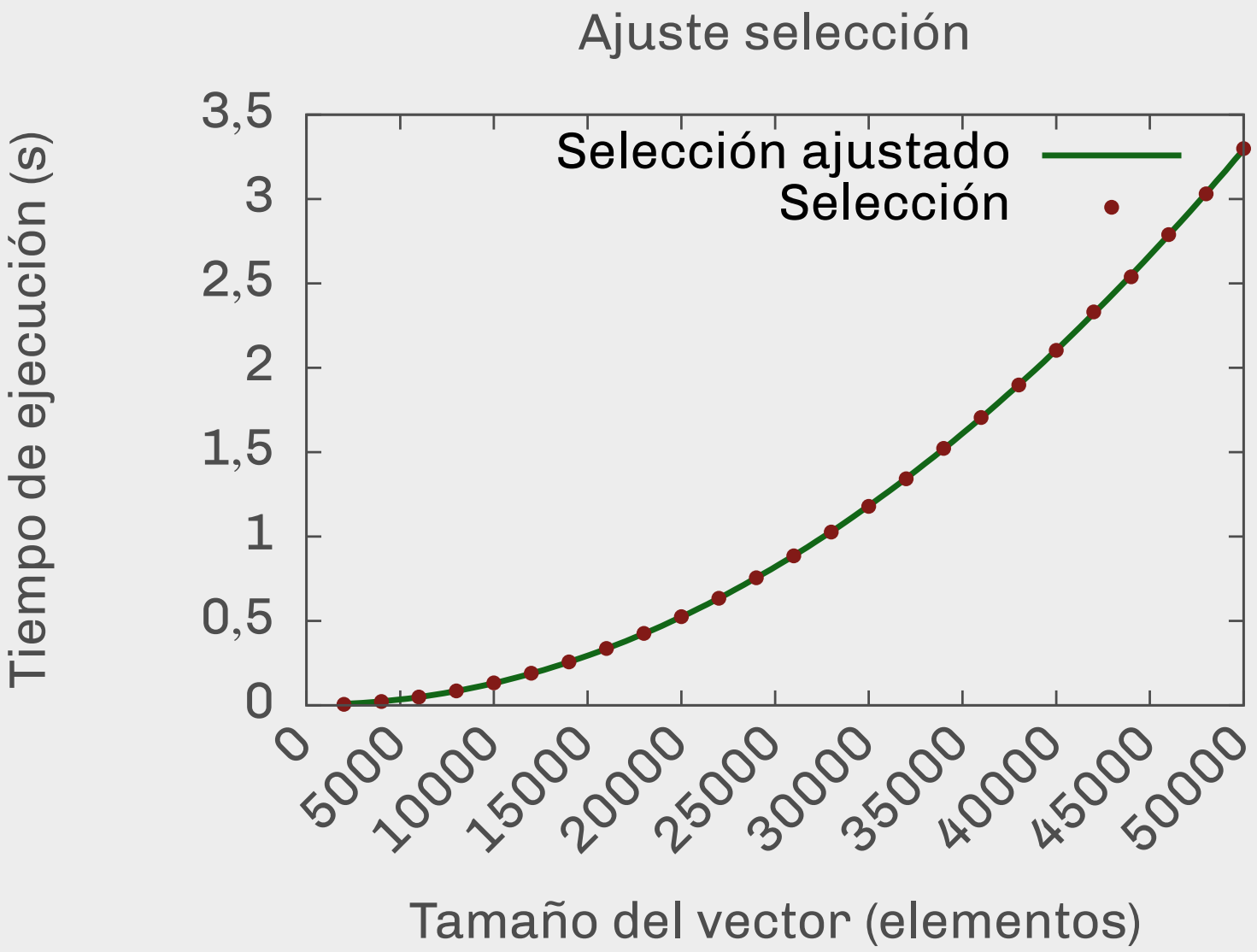

Algoritmos de ordenación $O(n^2)$
Tiempo de ejecución (s)
Tamaño del vector (elementos)
Burbuja
Inserción
Selección
0
1
2
3
4
5
6
7
8
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000

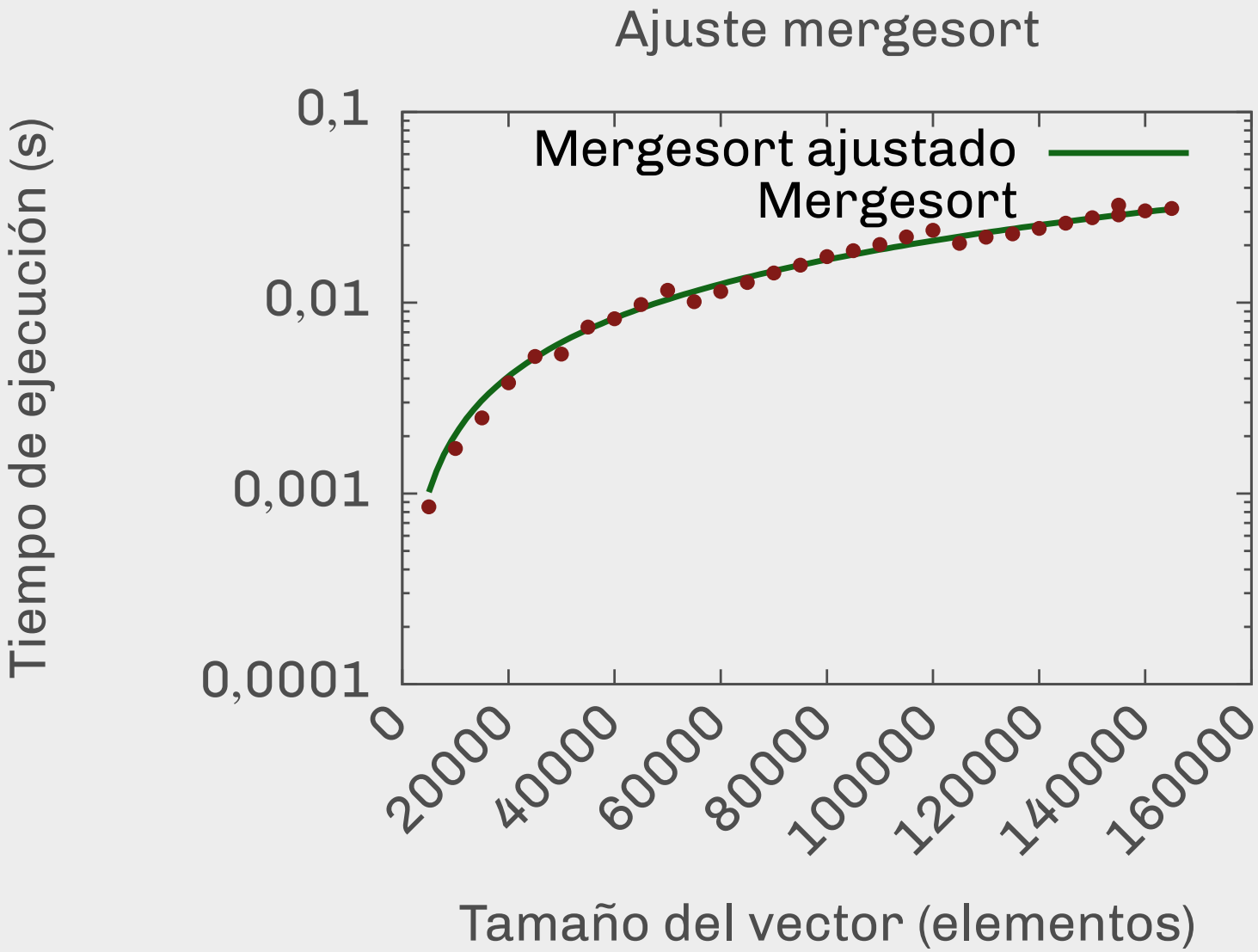

Algoritmos de ordenación O(nlogn)
Tiempo de ejecución (s)
0,035
0,03
0,025
0,02
0,015
0,01
0,005
0
Mergesort
Heapsort
Quicksort
0
20000
40000
60000
80000
100000
120000
140000
160000
Tamaño del vector (elementos)

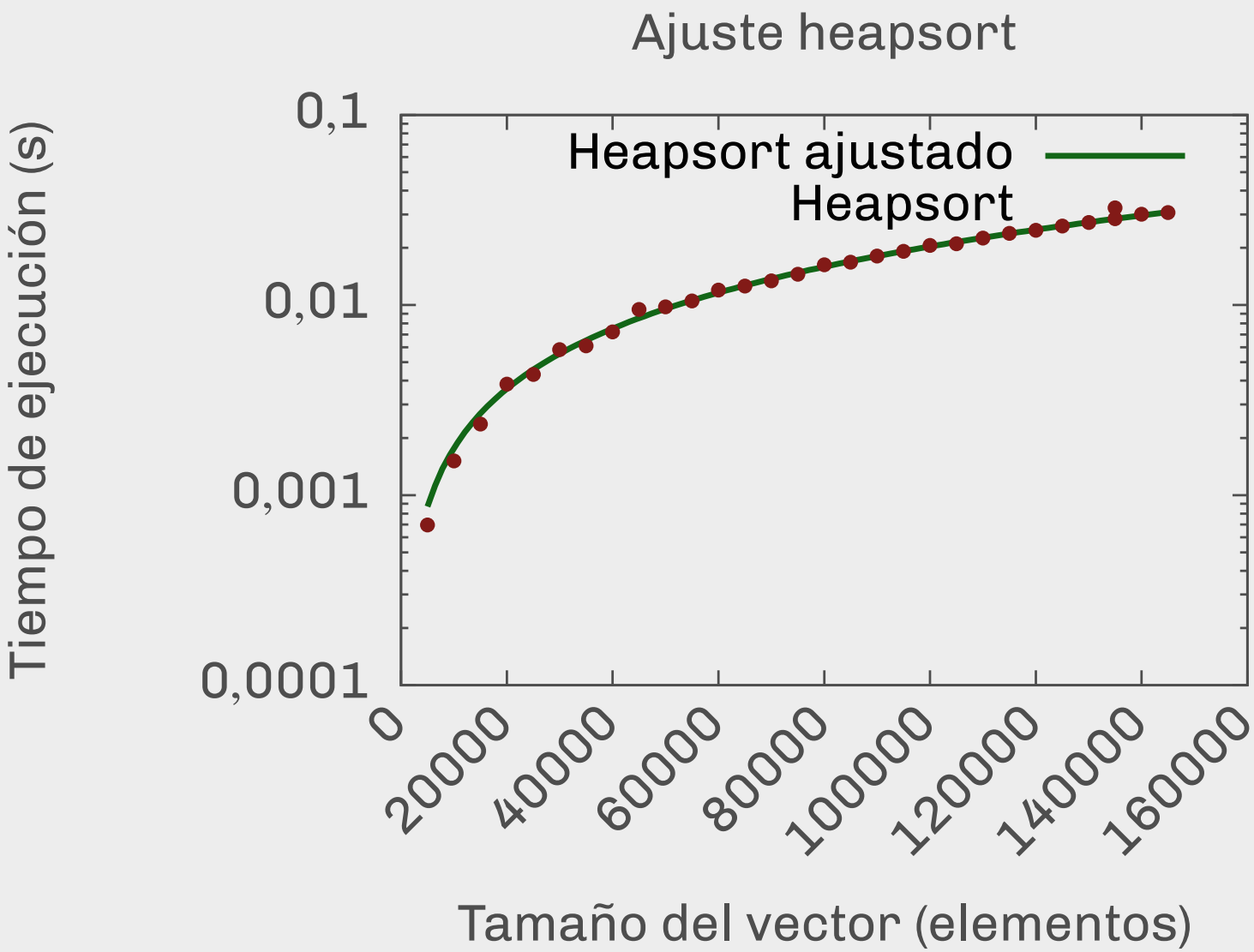

Algoritmos de ordenación
Tiempo de ejecución (s)
10
1
0,1
0,01
0,001
0,0001
0
20000
40000
60000
80000
100000
120000
140000
160000
Tamaño del vector (elementos)
Mergesort
Heapsort
Quicksort
Burbuja
Inserción
Selección

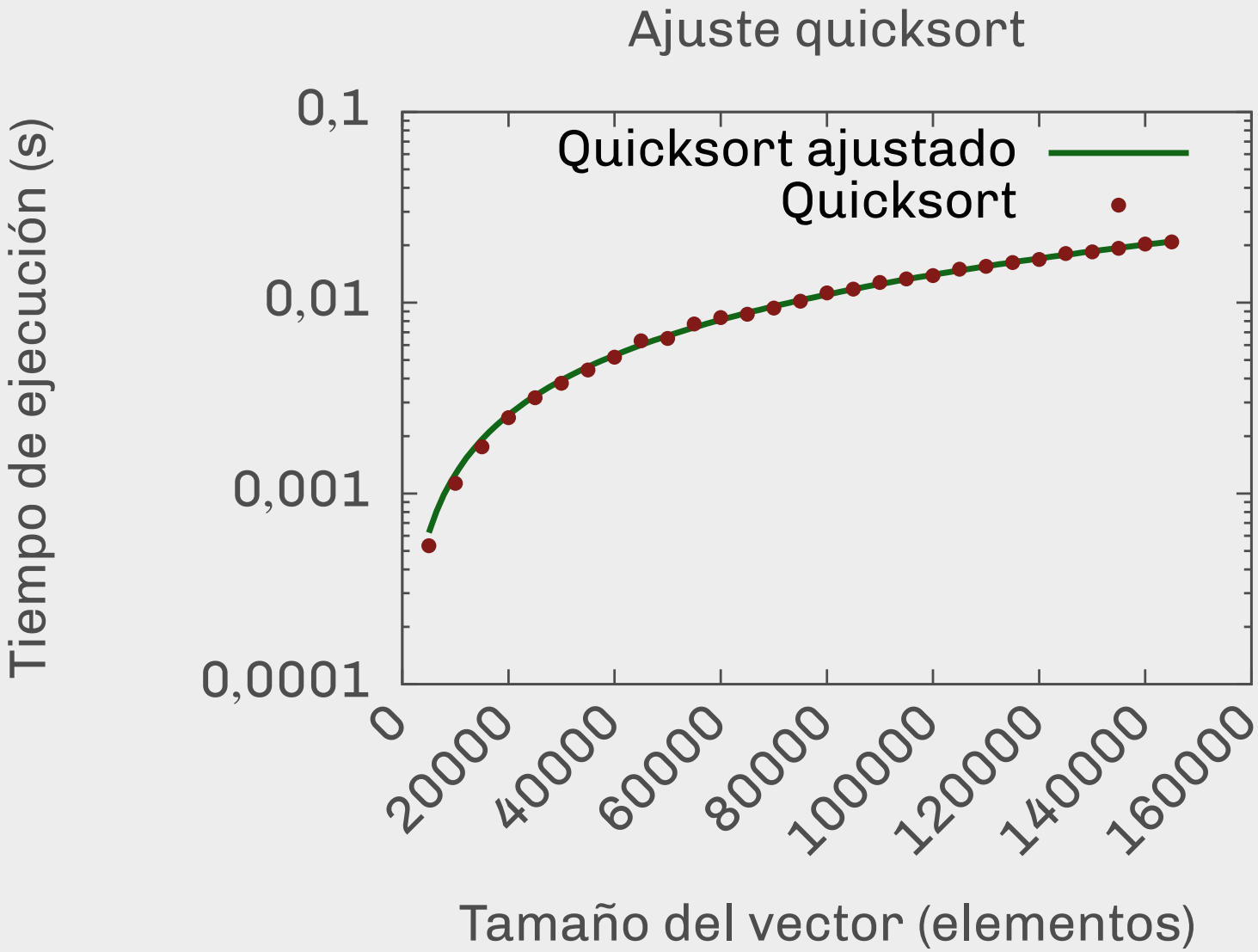

Ajuste burbuja
Burbuja ajustada
Burbuja
Tiempo de ejecución (s)
Tamaño del vector (elementos)
0
1
2
3
4
5
6
7
8
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000

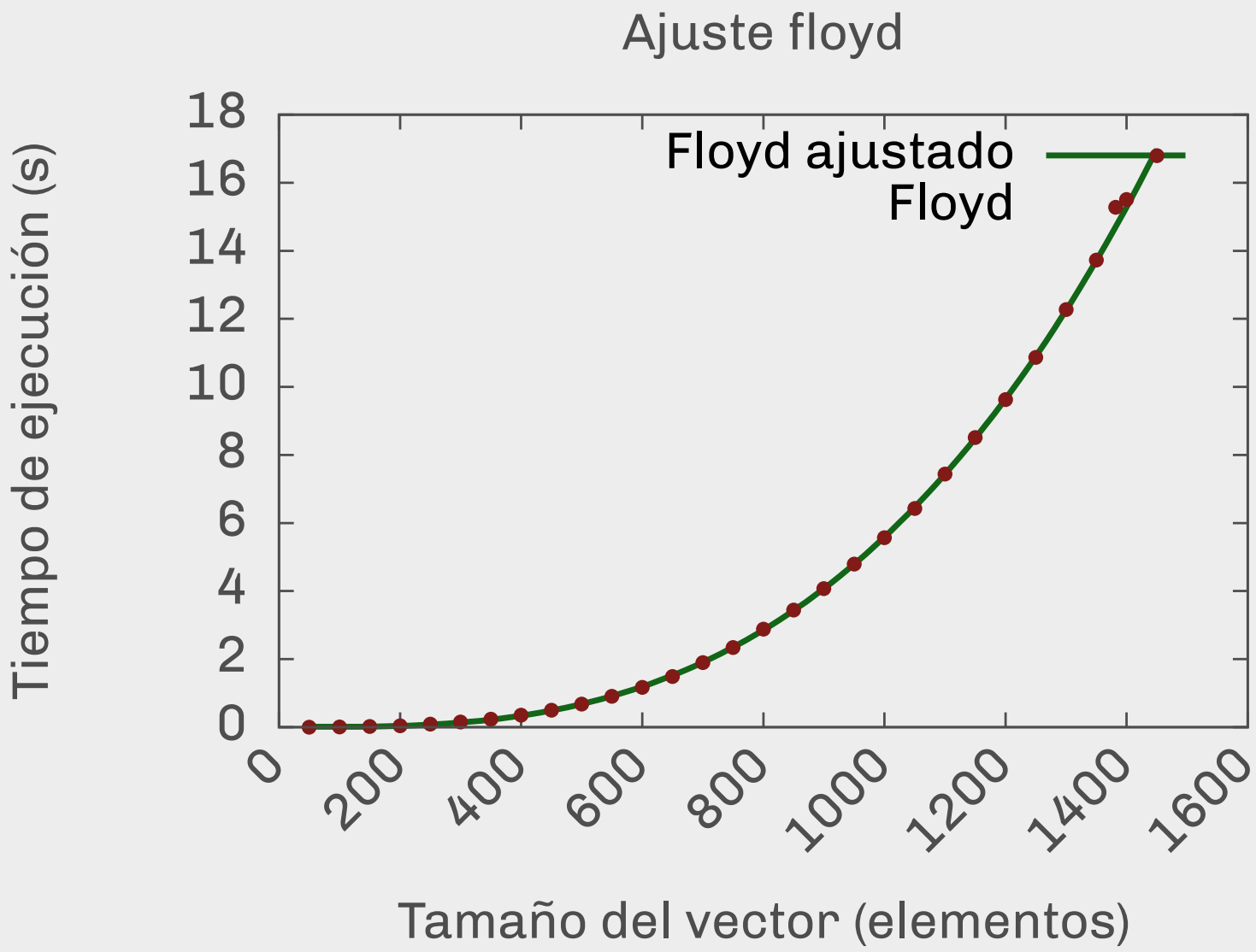

Ajuste inserción
Inserción ajustado
Inserción
Tiempo de ejecución (s)
Tamaño del vector (elementos)
0
0,5
1
1,5
2
2,5
3
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000

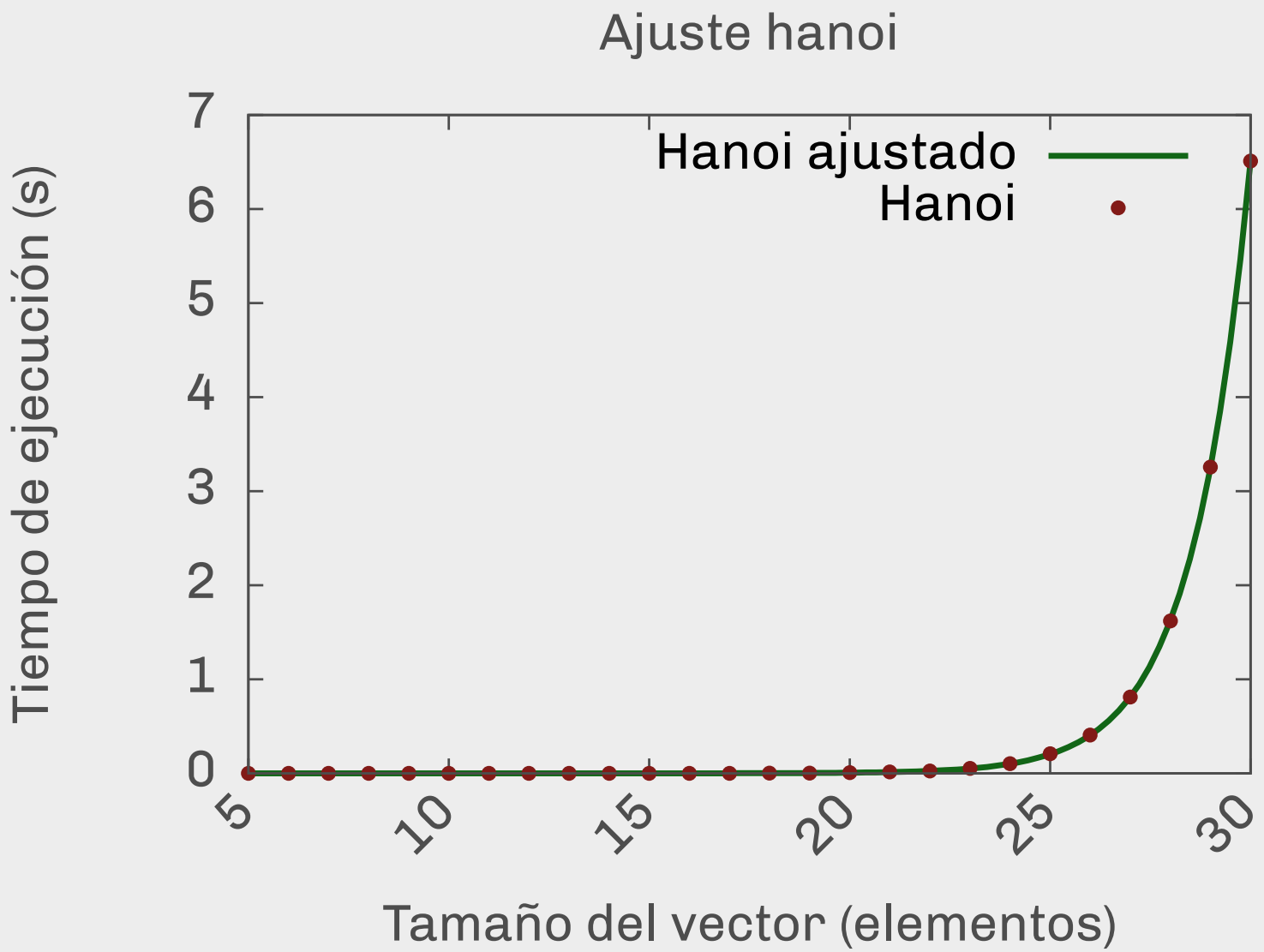

Ajuste selección
Selección ajustado
Selección
Tiempo de ejecución (s)
Tamaño del vector (elementos)
0
0,5
1
1,5
2
2,5
3
3,5
0
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000


Ajuste mergesort
Mergesort ajustado
Mergesort
Tiempo de ejecución (s)
0,1
0,01
0,001
0,0001
0
20000
40000
60000
80000
100000
120000
140000
160000
Tamaño del vector (elementos)


Ajuste heapsort
Heapsort ajustado
Heapsort
Tiempo de ejecución (s)
0,1
0,01
0,001
0,0001
0
20000
40000
60000
80000
100000
120000
140000
160000
Tamaño del vector (elementos)


Ajuste quicksort
Quicksort ajustado
Quicksort
Tiempo de ejecución (s)
0,1
0,01
0,001
0,0001
0
20000
40000
60000
80000
100000
120000
140000
160000
Tamaño del vector (elementos)


Ajuste floyd
Floyd ajustado
Floyd
Tiempo de ejecución (s)
Tamaño del vector (elementos)
0
2
4
6
8
10
12
14
16
18
0
200
400
600
800
1000
1200
1400
1600


Ajuste hanoi
Hanoi ajustado
Hanoi
Tiempo de ejecución (s)
Tamaño del vector (elementos)
0
1
2
3
4
5
6
7
5
10
15
20
25
30

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 $O(n^2)$