Algorítmica: Práctica 1
Eficiencia de algoritmos

Por:
Jesús Sánchez de Lechina Tejada
Miguel Ángel Robles Urquiza
Antonio Martín Ruíz
Álvaro López Jiménez

ÍNDICE

ENUNCIADO

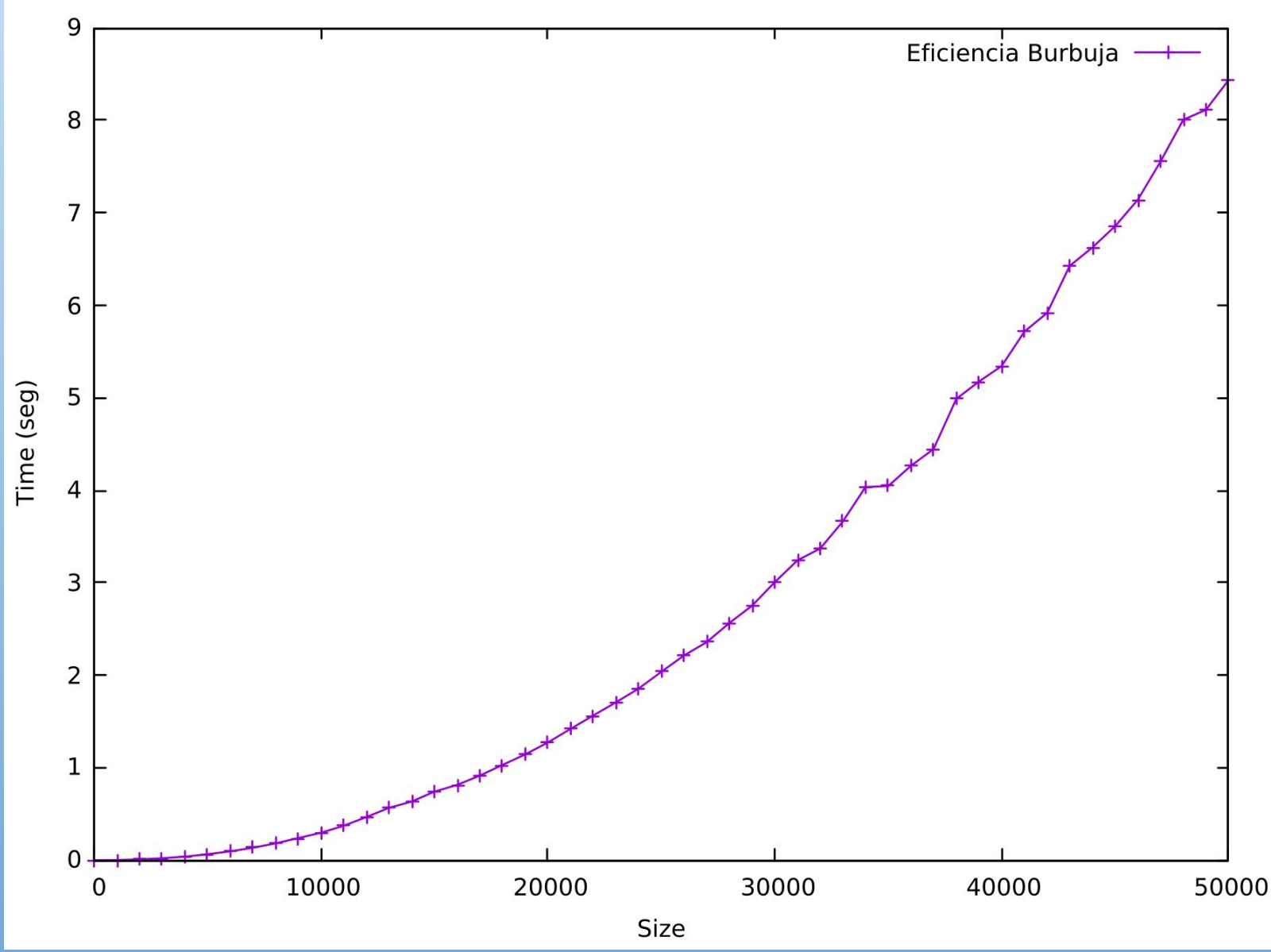
1. Calcule la eficiencia empírica, siguiendo las indicaciones de la sección 3. Defina adecuadamente los tamaños de entrada de forma tal que se generen al menos 25 datos. Incluya en la memoria tablas diferentes para los algoritmos de distinto orden de eficiencia (una con los algoritmos de orden O(n2), otra con los O(nlogn), otra con O(n3) y otra con O(2n)).

2. 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 grafica con todos ellos, para poder apreciar las diferencias en rendimiento de algoritmos con diferente orden de eficiencia.

BURBUJA

- Algoritmo para ordenar un vector
- De orden $O(n^2)$
- El número de comparaciones no depende del orden de los términos, si no del número de términos

BURBUJA


Eficiencia Burbuja
Time (seg)
Size
0
1
2
3
4
5
6
7
8
9
0
10000
20000
30000
40000
50000

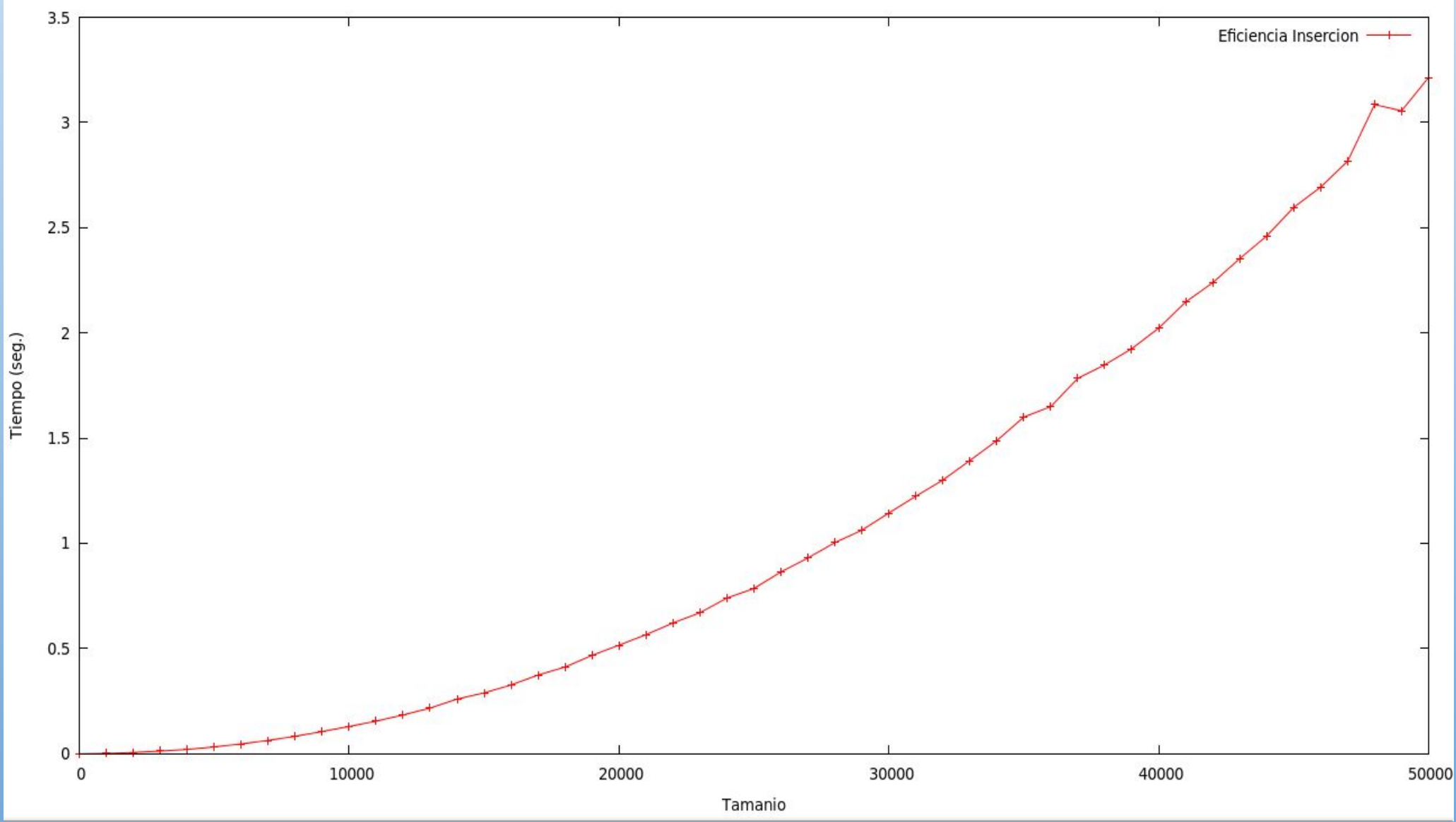
BURBUJA

TAMAÑO	TIEMPO (SEG.)
0	0
5000	0.063669
10000	0.296445
15000	0.740756
20000	1.27296
25000	2.03655
30000	3.00578
35000	4.04552
40000	5.34389
45000	6.85601
50000	8.43534

INSERCIÓN

- Algoritmo para ordenar un vector
- De orden $O(n^2)$
- Bueno para tamaños muy pequeños

INSERCIÓN


Eficiencia Insercion
Tiempo (seg.)
Tamanio
0
0.5
1
1.5
2
2.5
3
3.5
0
10000
20000
30000
40000
50000

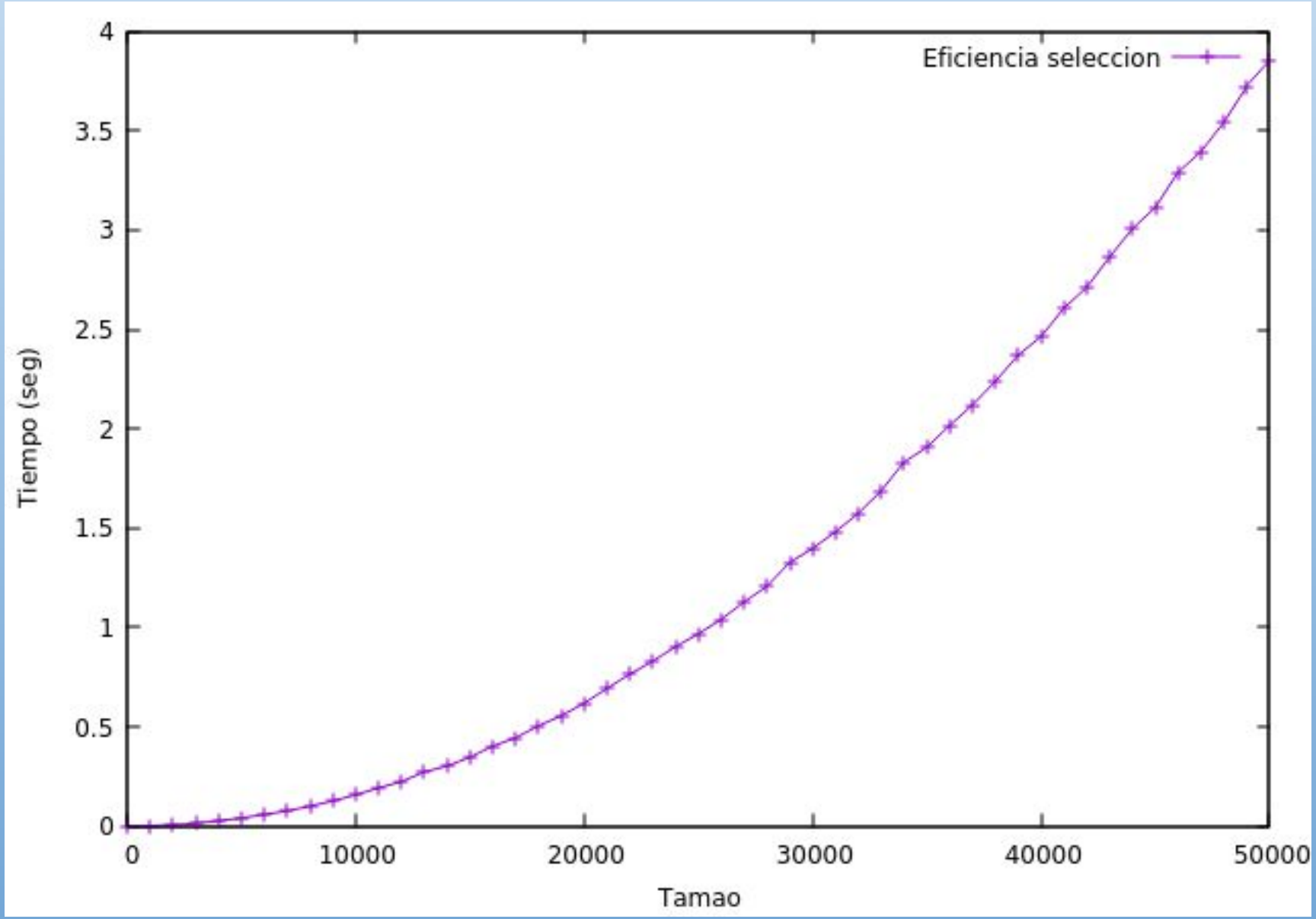
INSERCIÓN

TAMAÑO	TIEMPO (SEG.)
0	1e-06
5000	0.032399
10000	0.129617
15000	0.289151
20000	0.51575
25000	0.784994
30000	1.14407
35000	1.59976
40000	2.02232
45000	2.59602
50000	3.21349

SELECCIÓN

- Algoritmo para ordenar un vector
- De orden $O(n^2)$
- En cada iteración busca de entre los elementos no ordenador del vector el menor y lo pone el primero de los no ordenados

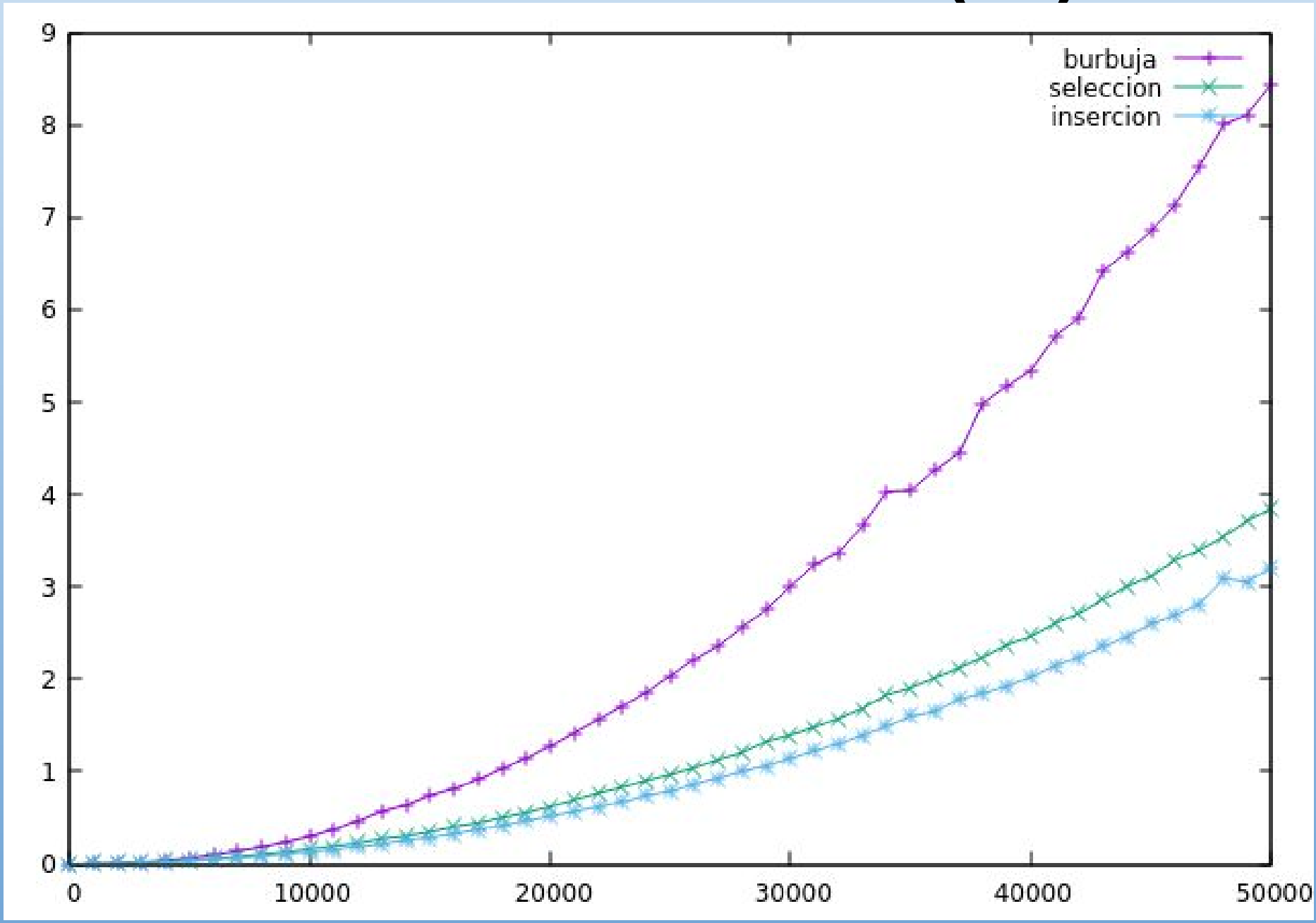
SELECCIÓN


Eficiencia seleccion
Tiempo (seg)
Tamao
0
0.5
1
1.5
2
2.5
3
3.5
4
0
10000
20000
30000
40000
50000

SELECCIÓN

TAMAÑO	TIEMPO (SEG.)
0	0
5000	0.041315
10000	0.157276
15000	0.346635
20000	0.618515
25000	0.967005
30000	1.39705
35000	1.90573
40000	2.46294
45000	3.11329
50000	3.84767

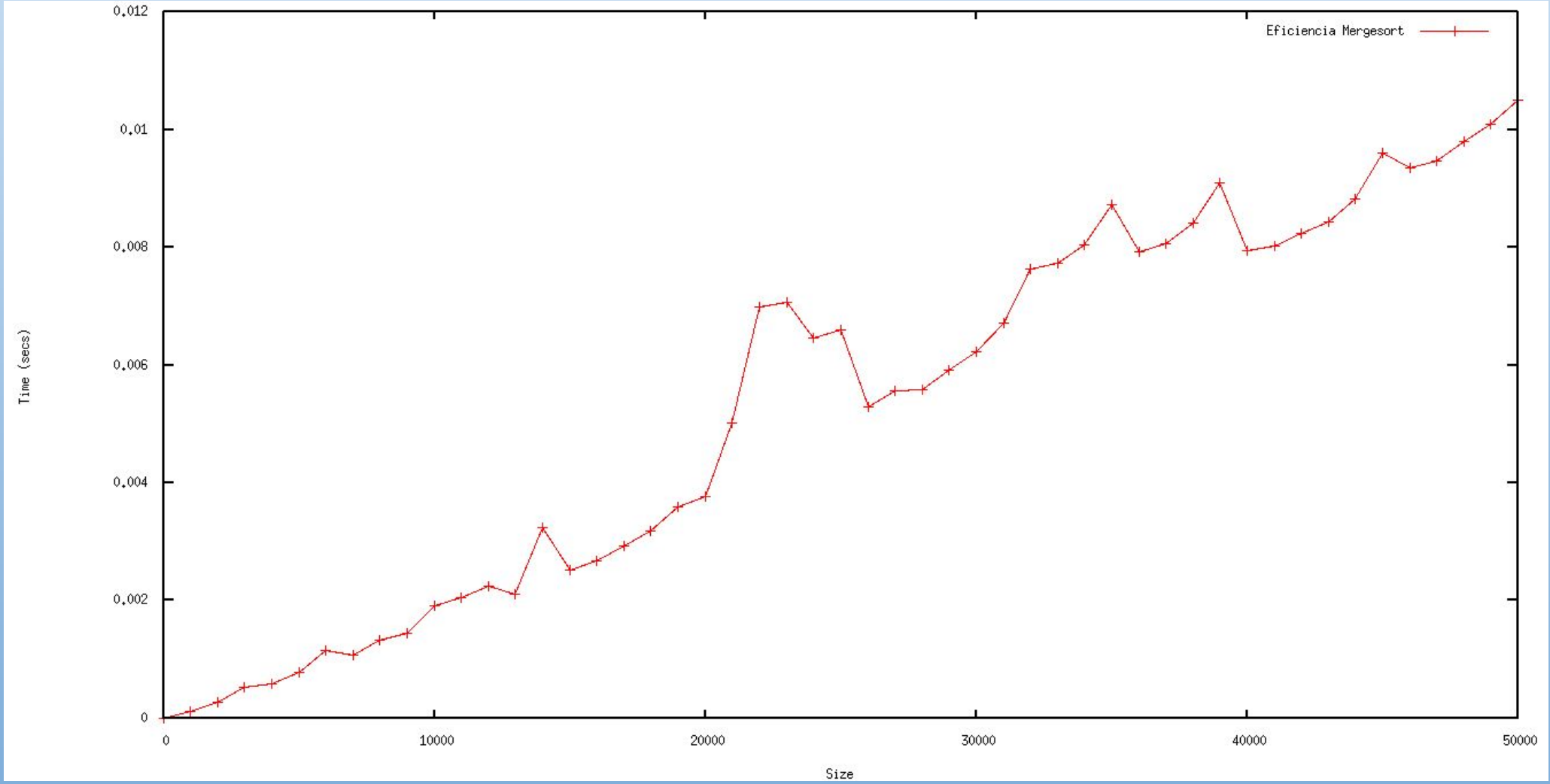
ALGORITMOS O(n²)



MERGESORT

- Algoritmo de ordenación de vectores
  - De orden O(nlogn)
  - Estrategia “Divide y vencerás”

MERGESORT


Eficiencia Mergesort
Time (secs)
Size
0.012
0.01
0.008
0.006
0.004
0.002
0
0
10000
20000
30000
40000
50000

MERGESORT

TAMAÑO	TIEMPO (SEG.)
0	8·10^-9
1 000	0.000110994
5 000	0.000782538
10 000	0.00191565
15 000	0.002519
20 000	0.00376
30 000	0.006229
40 000	0.007935
45 000	0.009597
50 000	0.010494

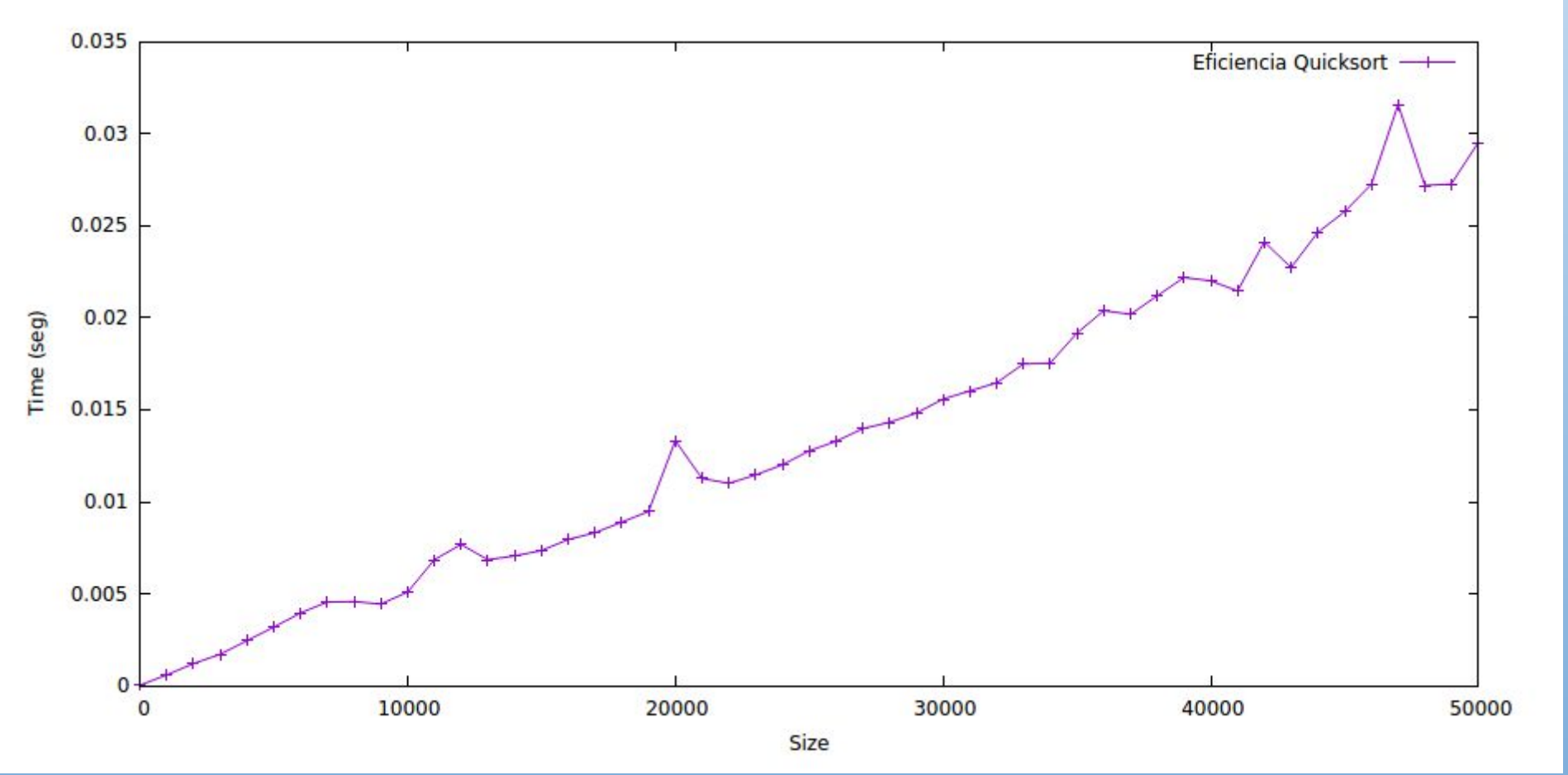
QUICKSORT

- Algoritmo para ordenar un vector

- De orden O(nlogn)

- El algoritmo básico del método Quicksort consiste en tomar cualquier elemento de la lista al cual denominaremos como pivote, dependiendo de la partición en que se elija, el algoritmo será más o menos eficiente

QUICKSORT


Eficiencia Quicksort
Time (seg)
Size
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
10000
20000
30000
40000
50000

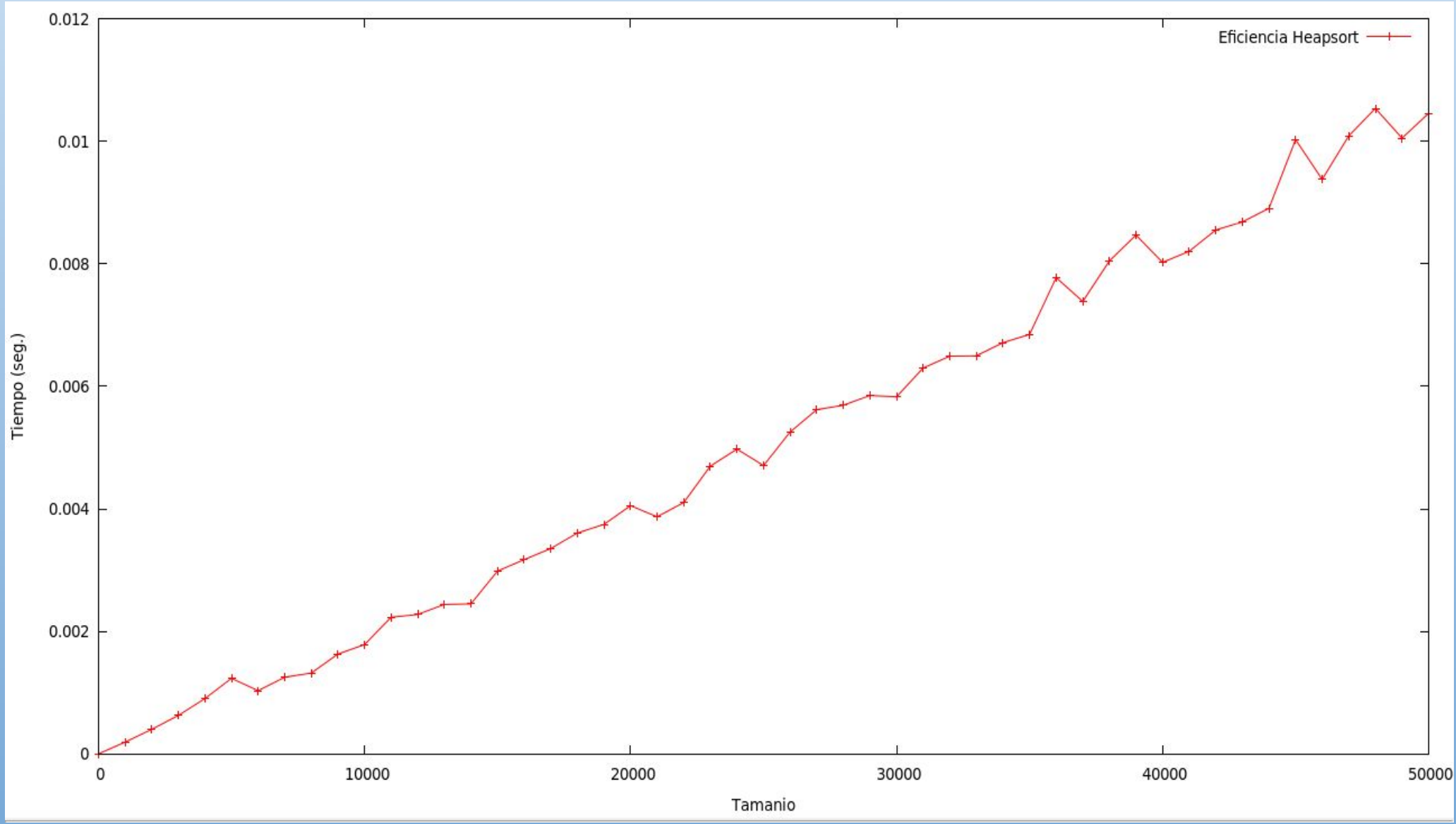
QUICKSORT

TAMAÑO	TIEMPO (SEG.)
0	1e-06
5000	0.003185
10000	0.005087
15000	0.007348
20000	0.013318
25000	0.012763
30000	0.015577
35000	0.019164
40000	0.022003
45000	0.025764
50000	0.029482

HEAPSORT

- Algoritmo para ordenar un vector
- De orden O(nlogn)
- Generalmente muy buen algoritmo (rápido)

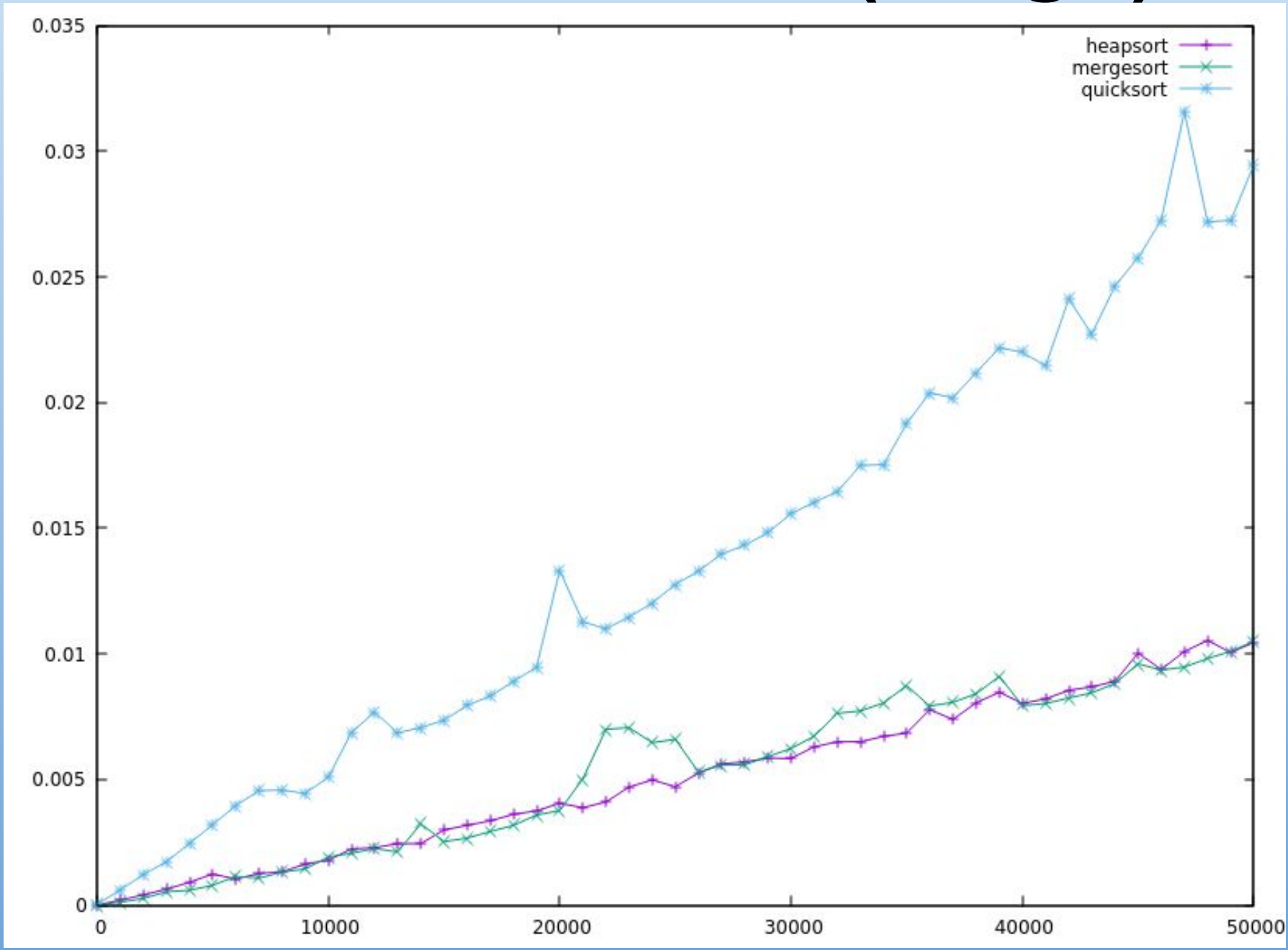
HEAPSORT


Eficiencia Heapsort
Tiempo (seg.)
Tamanio
0.012
0.01
0.008
0.006
0.004
0.002
0
0
10000
20000
30000
40000
50000

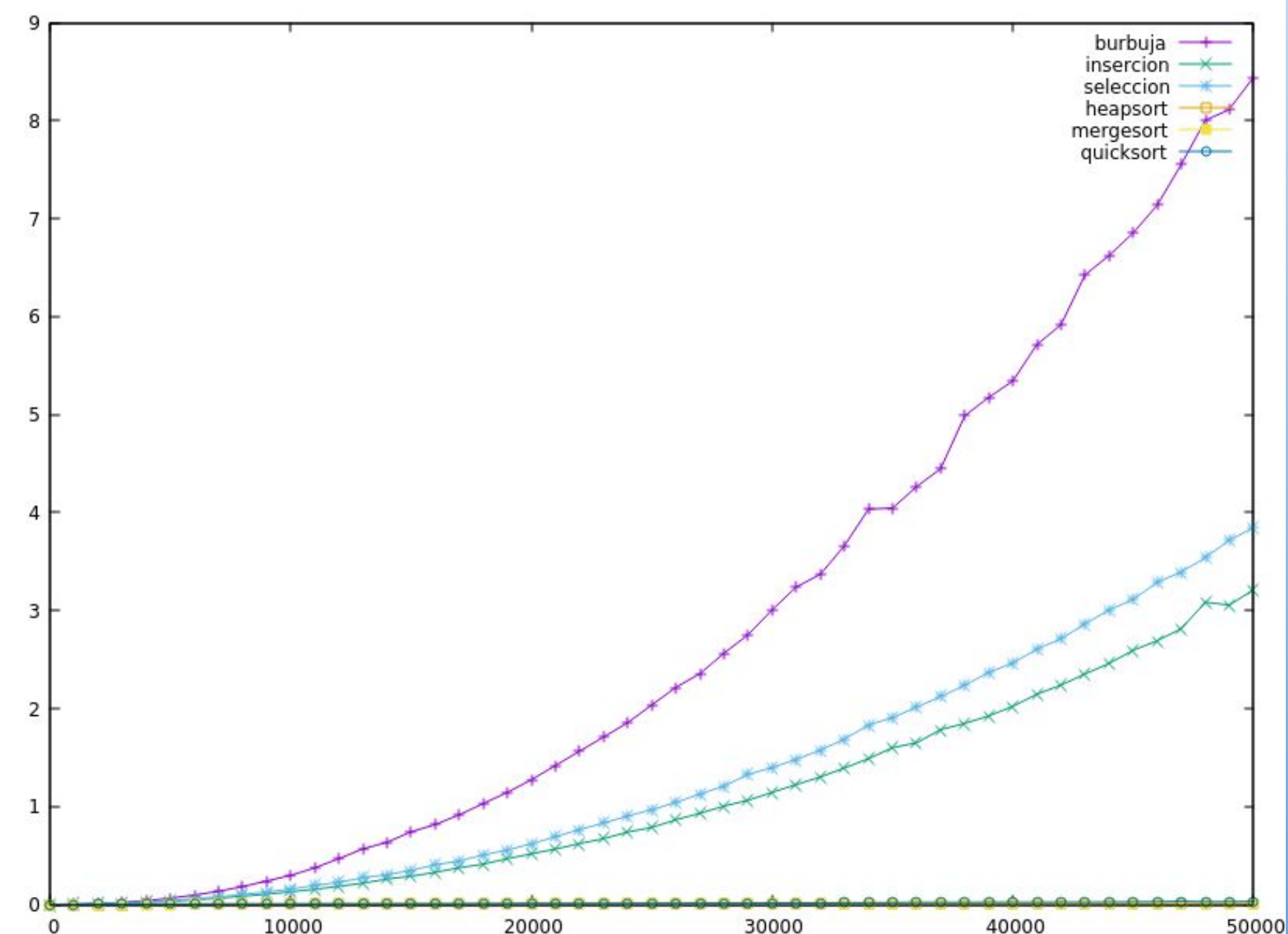
HEAPSORT

TAMAÑO	TIEMPO (SEG.)
0	1e-06
5000	0.001234
10000	0.001784
15000	0.002985
20000	0.004051
25000	0.004707
30000	0.00583
35000	0.006847
40000	0.008022
45000	0.010026
50000	0.010453

ALGORITMOS O(nlogn)



ALGORITMOS DE ORDENACIÓN



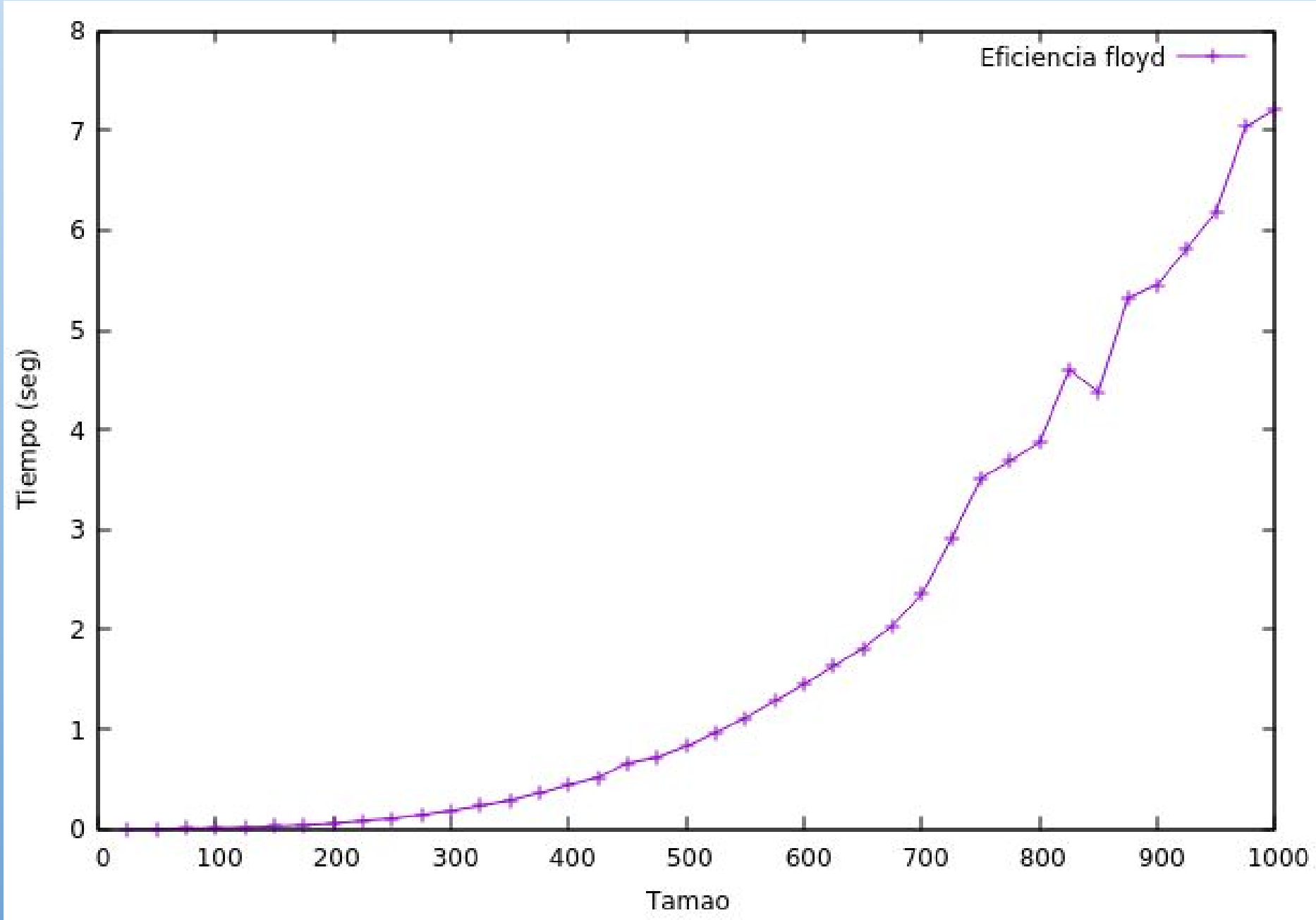
FLOYD

- Algoritmo para encontrar el camino mínimo en grafos dirigidos ponderados

- De orden $O(n^3)$

- Compara todos los caminos posibles entre cada par de vértices del grafo y luego elige el menor camino que los recorre a todos

FLOYD


Eficiencia floyd
Tiempo (seg)
Tamao
0
1
2
3
4
5
6
7
8
0
100
200
300
400
500
600
700
800
900
1000

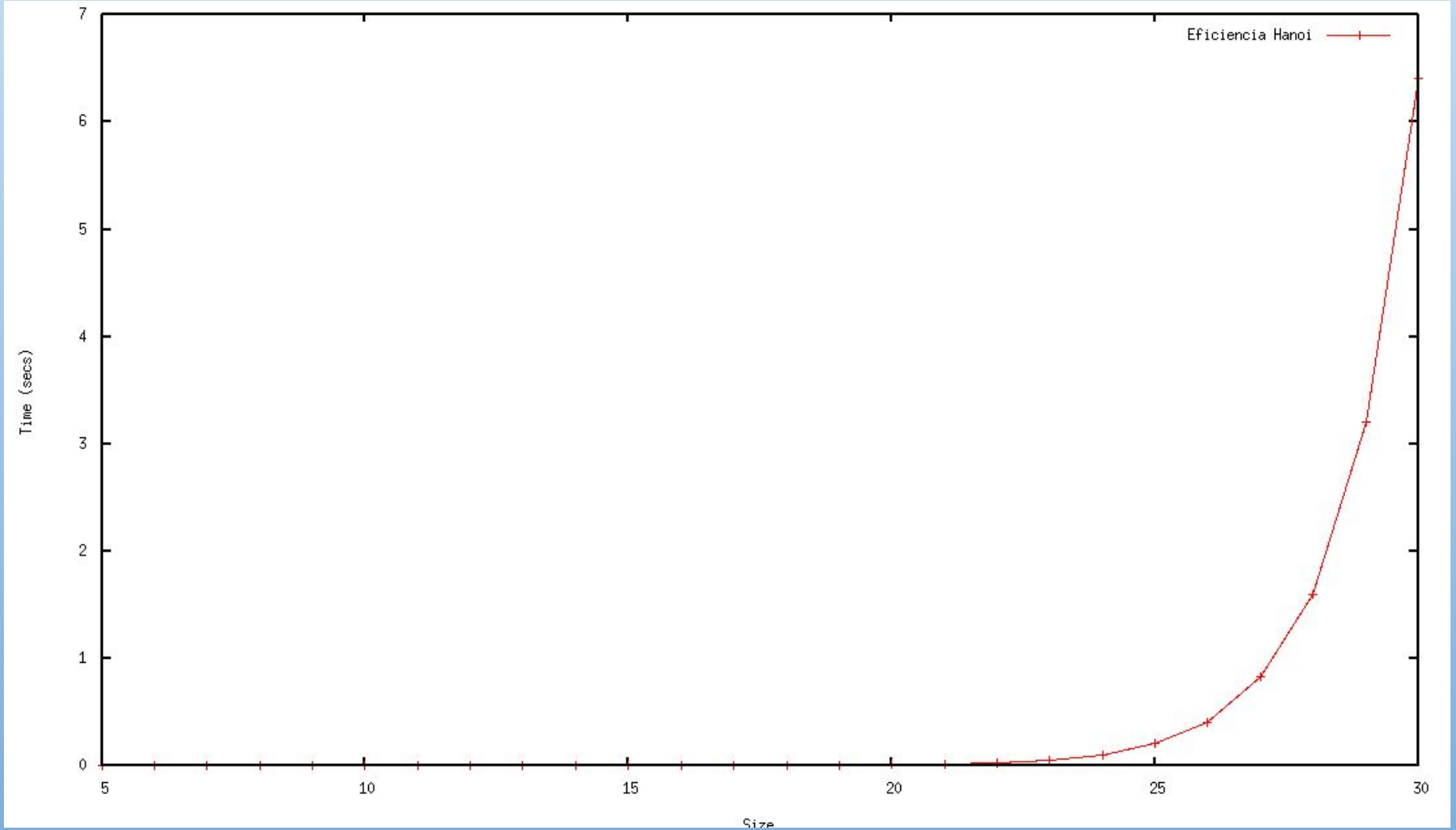
FLOYD

TAMAÑO	TIEMPO (SEG.)
0	0
100	0.009614
200	0.055406
300	0.184749
400	0.444626
500	0.827658
600	1.45198
700	2.35723
800	3.87428
900	5.45157
1000	7.20832

HANOI

- Algoritmo para resolver el problema de las torres de hanoi
- De orden O(2^n)
- Inviable para tamaños elevados

HANOI



HANOI

TAMAÑO	TIEMPO (en segundos)
5	2e-06
6	3e-06
8	5e-06
10	1e-05
15	0.000273
25	0.202328
28	1.59081
29	3.20046
30	6.41383