

PRÁCTICA 1: EFICIENCIA

Pablo Moreno Megías, Diego Lerena García, Manuel Vallejo Felipe,
Ángel Díaz de la Torre, Francisco Navarro Morales, Marcel Kemp
Muñoz y David Redondo Correa

Universidad de Granada

22 de marzo de 2017

1 Eficiencia empírica

- Algoritmos de orden $n \log n$.
- Algoritmos de orden n^2 .
- Algoritmo de orden n^3 .
- Algoritmo de orden 2^n .

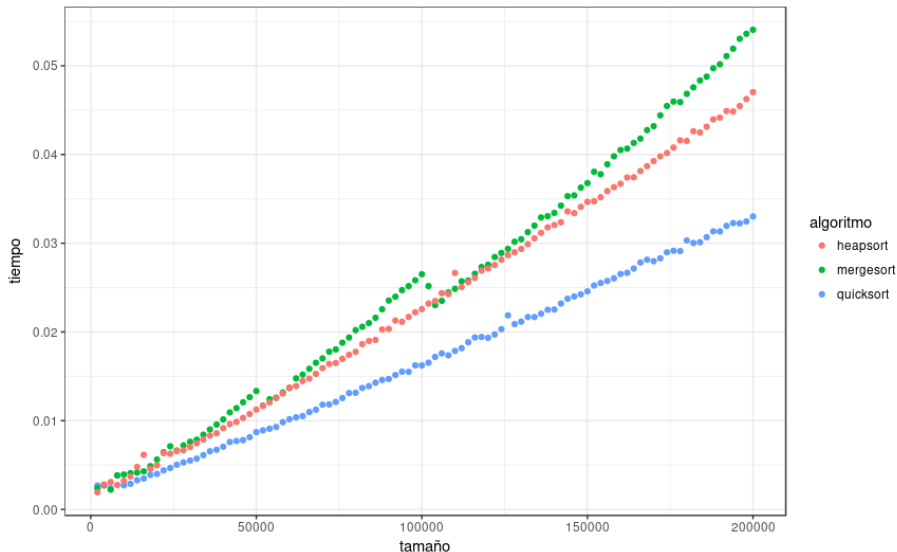
2 Eficiencia híbrida.

- Algoritmos de orden $n \log n$.
- Algoritmos de orden n^2 .
- Algoritmo de orden n^3 .
- Algoritmo de orden 2^n .

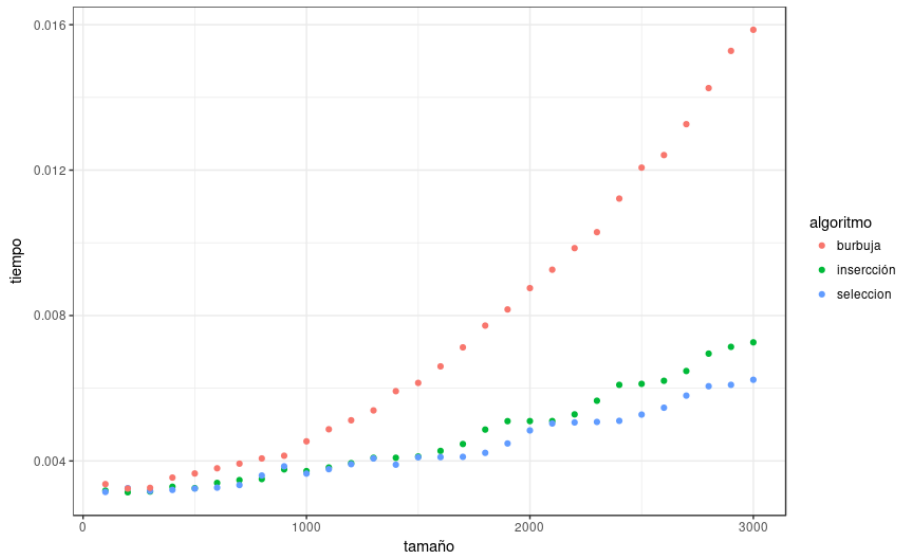
3 Otros análisis

- Burbuja con distintas optimizaciones.
- Comparación del Heapsort en distintos ordenadores.
- Mal ajuste de la función de Hanoi
- Comparación de Heapsort y Quicksort en el peor de los casos del Quicksort

Algoritmos de orden $n \log n$

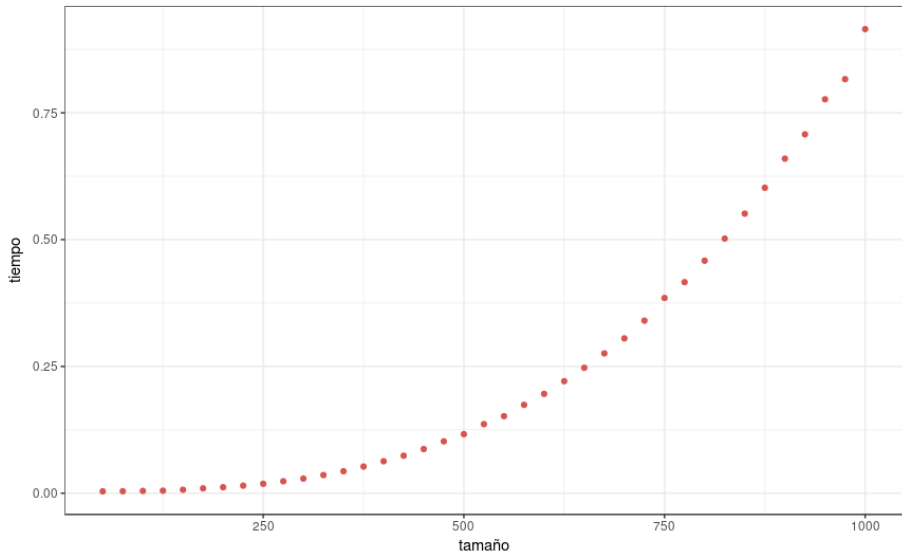


Algoritmos de orden n^2 .



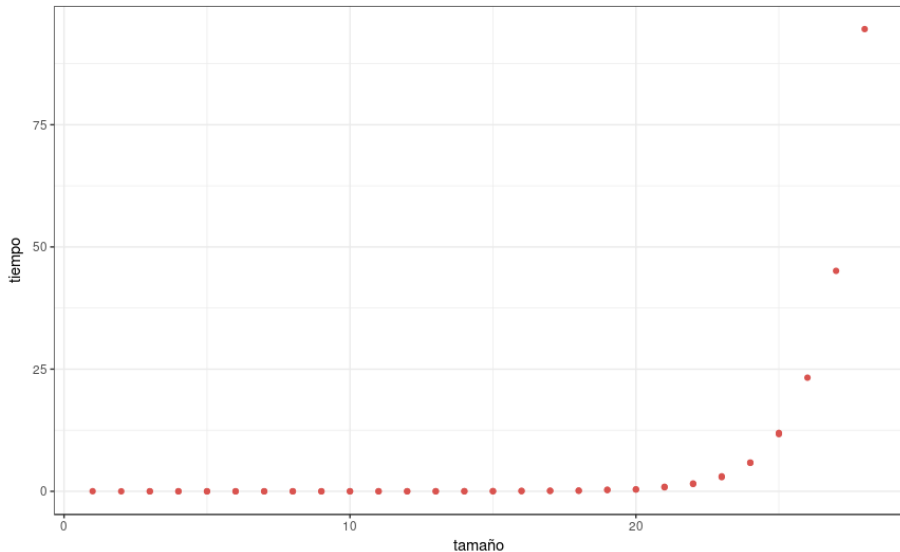
Algoritmo de orden n^3 .

Eficiencia empírica del algoritmo de Floyd



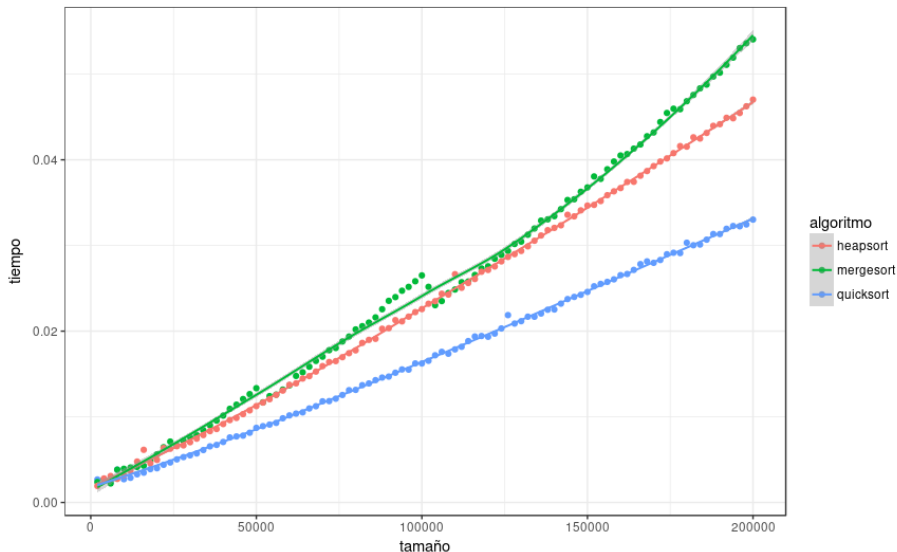
Algoritmo de orden 2^n .

Eficiencia empírica del algoritmo hanoi



Eficiencia híbrida

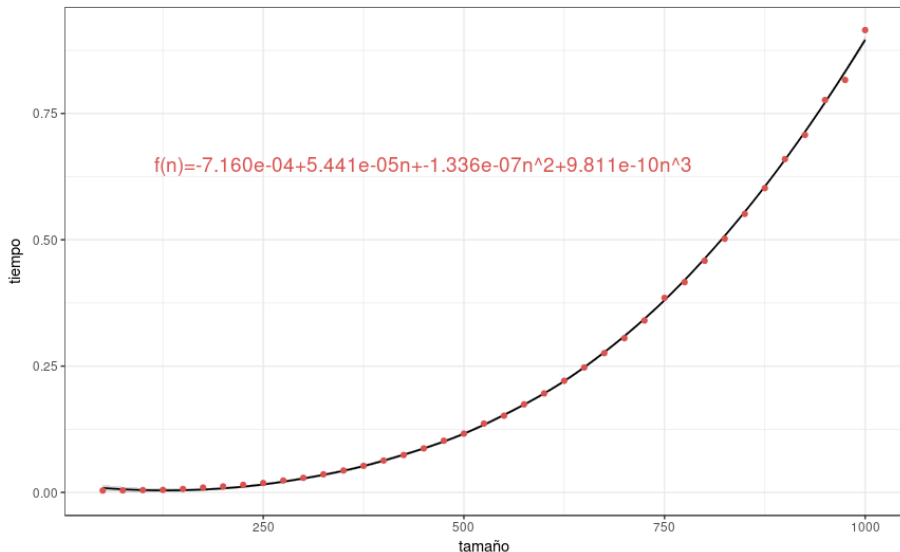
Comparación híbrida de algoritmos de ordenación



- Heapsort: $f(n) = 4,34068 * 10^5 * n \log(0,901865 * n) + 0,00511096$
- Quicksort: $f(n) = 3,02225 * 10^9 * n \log(0,901855 * n) + 0,00171908$
- Mergesort: $f(n) = 3,02225 * 10^9 * n \log(0,901855 * n) + 0,00171908$

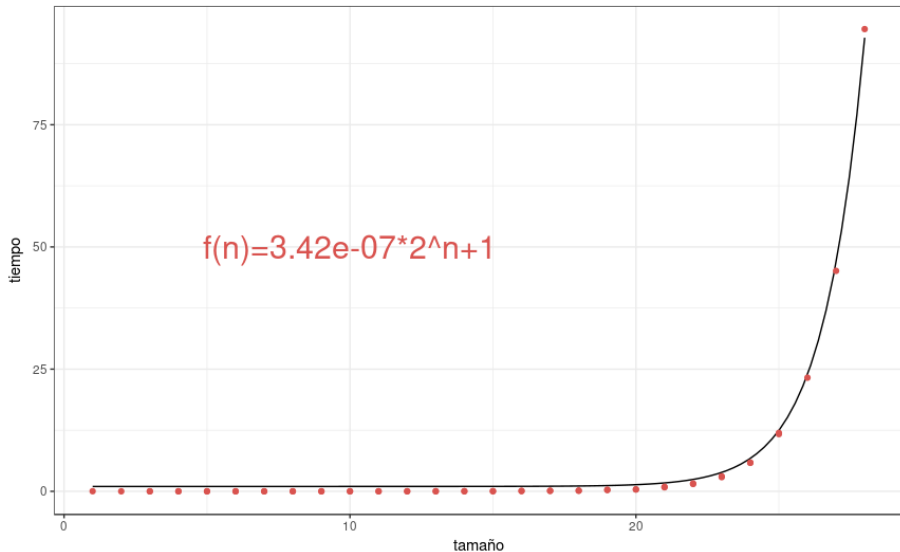
Algoritmo de Floyd

Eficiencia híbrida del algoritmo de Floyd



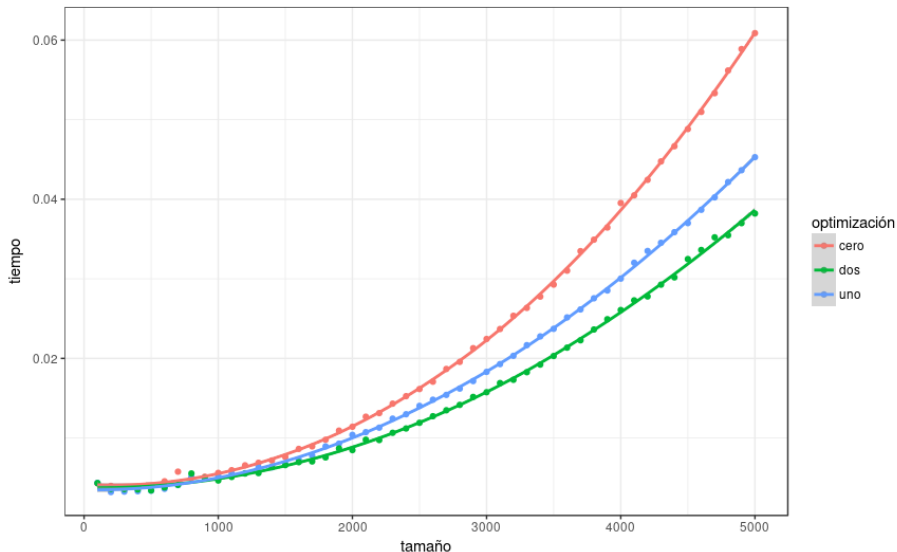
Algoritmo de Hanoi

Eficiencia híbrida del algoritmo hanoi

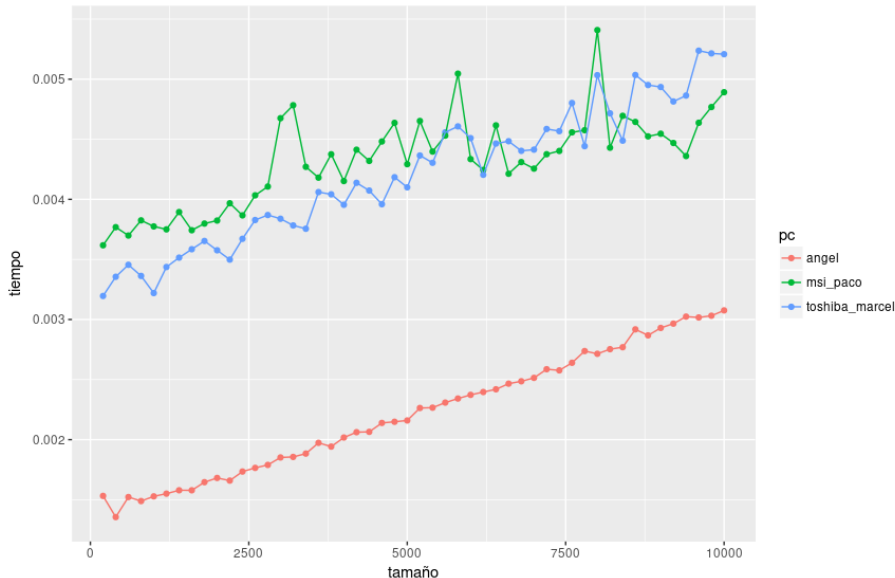


Burbuja con distintas optimizaciones.

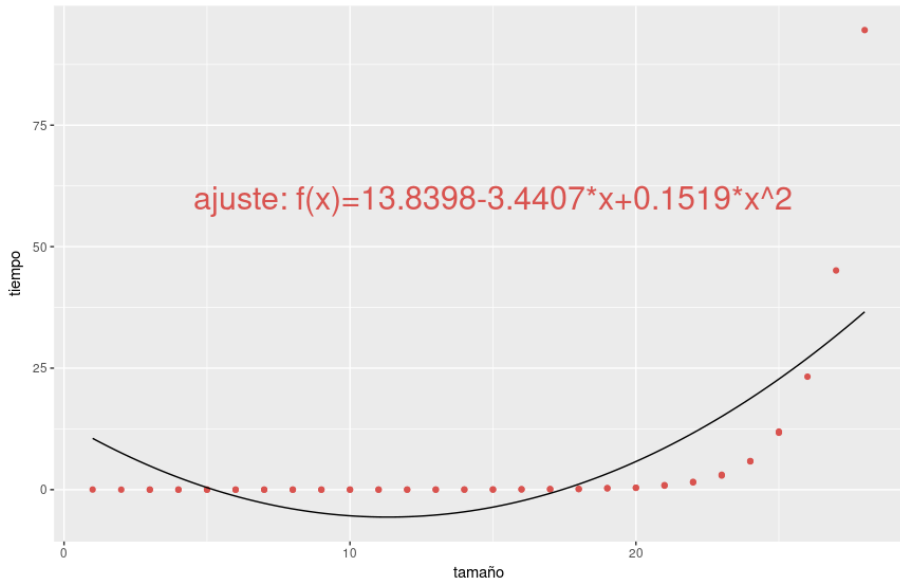
Comparación de tiempos según el nivel de optimización al compilar



Comparación del Heapsort en distintos ordenadores



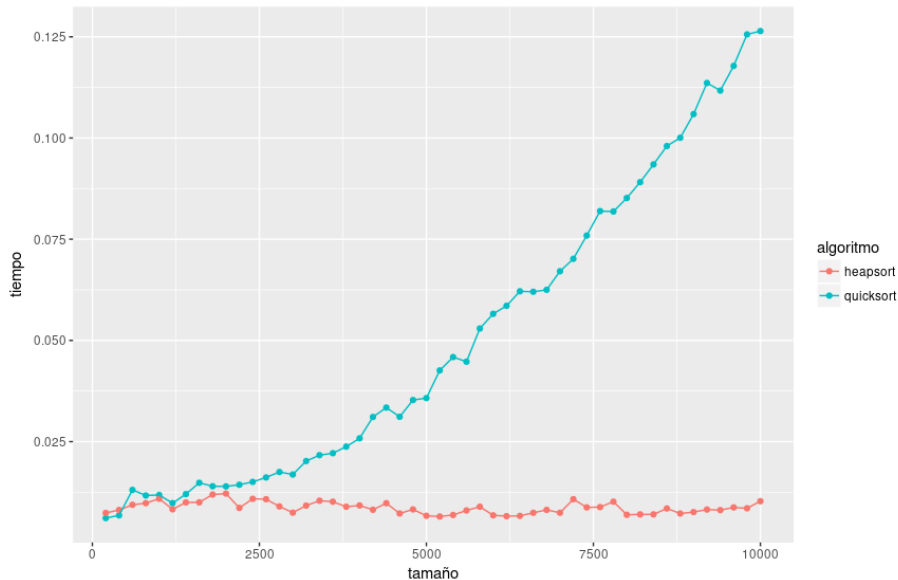
Mal ajuste de la función de Hanoi



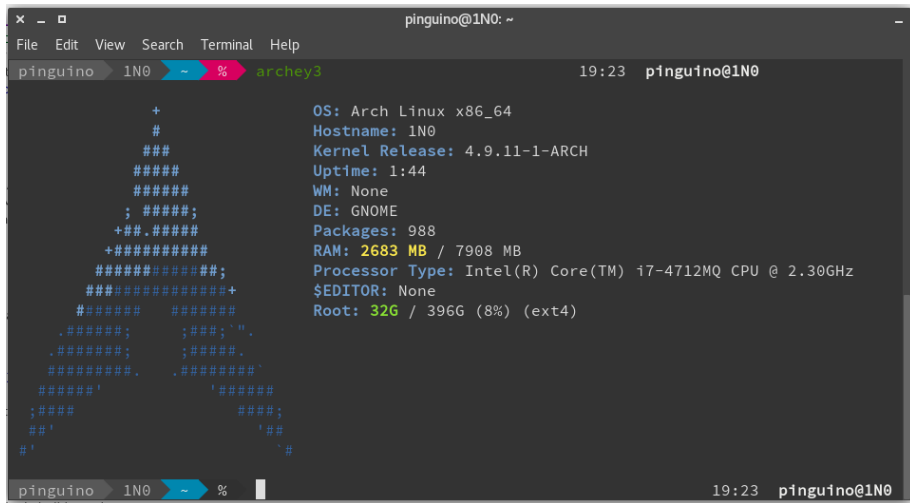
Heapsort y Quicksort [VECTOR DESORDENADO]



Heapsort y Quicksort [VECTOR ORDENADO]



COMPUTADOR UTILIZADO EN LAS MEDIDAS



A terminal window titled 'pinguino@1N0: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'pinguino@1N0' followed by a blue arrow pointing right and a pink arrow pointing left. The user has entered 'archey3'. The output shows system information: OS: Arch Linux x86_64, Hostname: 1N0, Kernel Release: 4.9.11-1-ARCH, Uptime: 1:44, WM: None, DE: GNOME, Packages: 988, RAM: 2683 MB / 7908 MB, Processor Type: Intel(R) Core(TM) i7-4712MQ CPU @ 2.30GHz, \$EDITOR: None, Root: 32G / 396G (8%) (ext4). On the left side of the terminal, there is a decorative ASCII art logo consisting of a grid of '#' characters forming a stylized shape.

```
pinguino@1N0: ~
File Edit View Search Terminal Help

pinguino@1N0 ~ % archey3

+ OS: Arch Linux x86_64
# Hostname: 1N0
### Kernel Release: 4.9.11-1-ARCH
#### Uptime: 1:44
##### WM: None
; #####; DE: GNOME
+##.##### Packages: 988
+##### RAM: 2683 MB / 7908 MB
#####; Processor Type: Intel(R) Core(TM) i7-4712MQ CPU @ 2.30GHz
#####+ $EDITOR: None
##### Root: 32G / 396G (8%) (ext4)

.#####; ;###;"
.#####; ;#####.
#####. #####`
#####' #####
;###' ###;
##' '#
#' '#`#

pinguino@1N0 ~ %
```

Figura: ARCH LINUX

The End