

## Práctica 1

### Algorítmica

#### Introducción

#### Ordenación

Burbuja

Selección

Inserción

#### Ordenación rápida

Quick Sort

Heap Sort

Merge Sort

#### Floyd

#### Hanoi

# Práctica 1: Eficiencia

## Algorítmica

Universidad de Granada

31 de mayo de 2016

# Índice

## Práctica 1

### Algorítmica

### Introducción

### Ordenación

Burbuja

Selección

Inserción

### Ordenación rápida

Quick Sort

Heap Sort

Merge Sort

### Floyd

### Hanoi

## 1 Introducción

## 2 Ordenación

- Burbuja
- Selección
- Inserción

## 3 Ordenación rápida

- Quick Sort
- Heap Sort
- Merge Sort

## 4 Floyd

## 5 Hanoi

# Introducción

## Práctica 1

### Algorítmica

#### Introducción

#### Ordenación

Burbuja  
Selección  
Inserción

#### Ordenación rápida

Quick Sort  
Heap Sort  
Merge Sort

#### Floyd

#### Hanoi

- El objetivo de ésta práctica es analizar eficiencias de forma empírica e híbrida.
- Para ello, hemos recogido los diferentes tiempos de los diferentes algoritmos que se ofrecían y los hemos comparado.
- En nuestro caso concreto, hemos utilizado la biblioteca de C++ más moderna y precisa destinado a obtener tiempos de reloj: la biblioteca **chrono**

# Especificaciones de la máquina

## Práctica 1

### Algorítmica

#### Introducción

#### Ordenación

Burbuja

Selección

Inserción

#### Ordenación rápida

Quick Sort

Heap Sort

Merge Sort

#### Floyd

#### Hanoi

- Procesador: Intel Core i5-3337U (2.7GHz x 2)
- Memoria RAM: 4GB
- Disco Duro: 500GB 5400 rpm
- SO: Manjaro Linux 15.2 Capella 64 bits

## Ordenación

## Práctica 1

## Algorítmica

## Ordenación

Floyd

Hanoi

Según hemos ido hemos estudiado, estos algoritmos que presentamos tienen teóricamente y calculando a partir del código una eficiencia de  $O(n^2)$ . Como esto es teórico, vamos a ver si efectivamente (o no) los algoritmos proporcionados se parecen a la gráfica de  $n^2$  recogiendo la información de 99 posibilidades distintas en cada algoritmo.

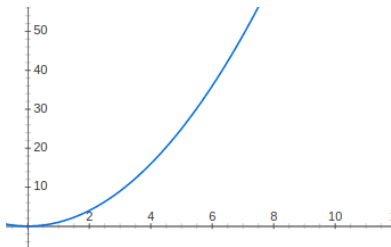


Figura: Gráfica de la función  $n^2$



# Burbuja

## Práctica 1

## Algorítmica

## Ordenación

## Burbuja

## Floyd

## Hanoi

La gráfica híbrida obtenida ha sido:

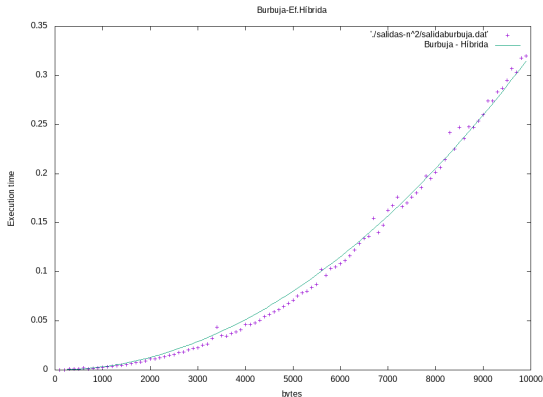


Figura: Ajuste híbrido algoritmo Burbuja y función  $n^2$

# Selección

## Práctica 1

## Algorítmica

## Ordenación

## Selección

Hanoi

La gráfica empírica obtenida ha sido:

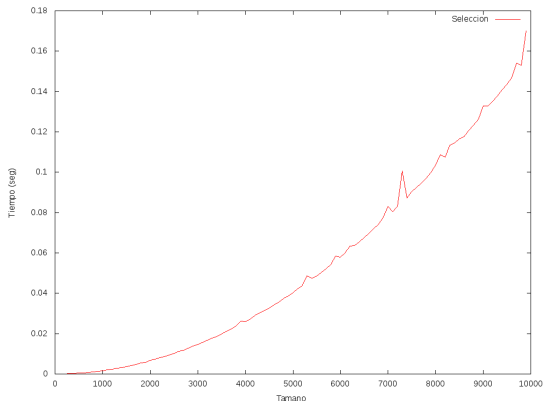


Figura: Algoritmo de selección, gráfica empírica.



# Selección

## Práctica 1

## Algorítmica

## Ordenación

## Selección

## Hanoi

La gráfica híbrida obtenida ha sido:

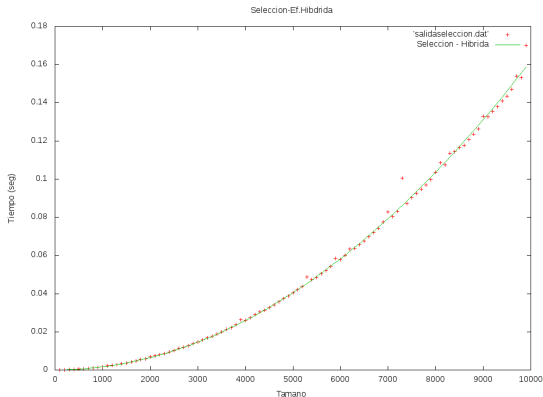


Figura: Gráfica ajuste híbrido. Selección y  $n^2$



# Inserción

Práctica 1

Algorítmica

Introducción

Ordenación

Burbuja

Selección

**Inserción**

Ordenación  
rápida

Quick Sort

Heap Sort

Merge Sort

Floyd

Hanoi

La gráfica híbrida obtenida ha sido:

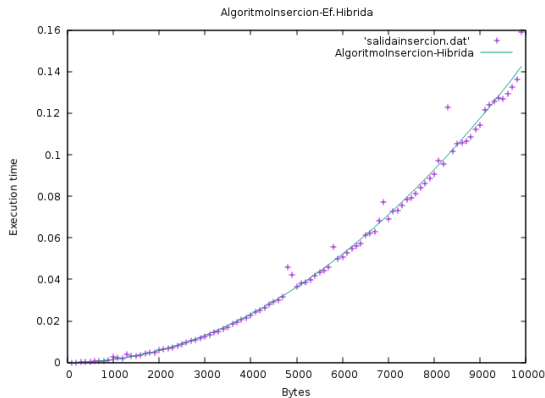


Figura: Gráfica ajuste híbrido. Inserción y  $n^2$

# Algoritmos $n^2$

## Práctica 1

### Algorítmica

#### Introducción

#### Ordenación

##### Burbuja

##### Selección

##### Inserción

#### Ordenación

##### rápida

##### Quick Sort

##### Heap Sort

##### Merge Sort

#### Floyd

#### Hanoi

Por último, mostramos el porcentaje de error así como las constantes ocultas obtenidas.

Algoritmo	Constante Oculta	Error
Burbuja	$a_0 = 3.20873e-09$	$+/- 1.403e-11$ (0.4372 %)
Selección	$a_0 = 1.61988e-09$	$+/- 4.818e-12$ (0.2975 %)
Inserción	$a_0 = 1.45151e-09$	$+/- 8.337e-12$ (0.5743 %)

# Ordenación rápida

Práctica 1

Algorítmica

Introducción

Ordenación

Burbuja

Selección

Inserción

Ordenación  
rápida

Quick Sort

Heap Sort

Merge Sort

Floyd

Hanoi

Según hemos ido estudiado, estos algoritmos que presentamos tienen teóricamente y calculando a partir del código una eficiencia de  $O(n * \log(n))$ . Como esto es teórico, vamos a ver si efectivamente (o no) los algoritmos proporcionados se parecen a la gráfica de  $n * \log(n)$  recogiendo la información de 99 posibilidades distintas en cada algoritmo.

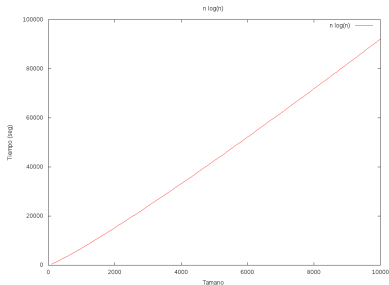


Figura: Gráfica de la función  $n * \log(n)$

# Quick Sort

Práctica 1

Algorítmica

Introducción

Ordenación

Burbuja

Selección

Inserción

Ordenación  
rápida

**Quick Sort**

Heap Sort

Merge Sort

Floyd

Hanoi

La gráfica empírica obtenida ha sido:

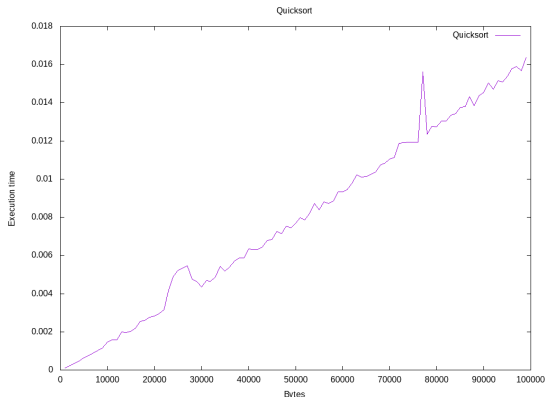


Figura: Algoritmo Quicksort, gráfica empírica.

# Quick Sort

Práctica 1

Algorítmica

Introducción

Ordenación

Burbuja

Selección

Inserción

Ordenación  
rápida

Quick Sort

Heap Sort

Merge Sort

Floyd

Hanoi

La gráfica híbrida obtenida ha sido:

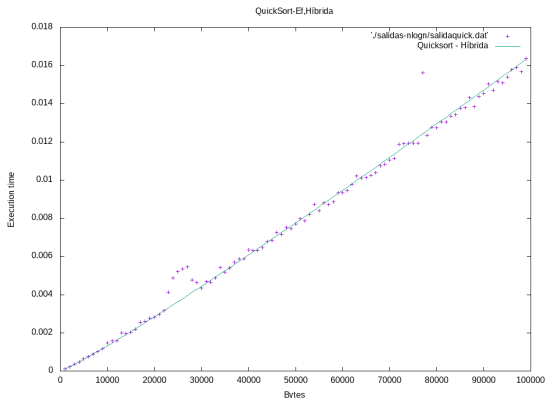


Figura: Gráfica híbrida Quicksort.

# Heap Sort

Práctica 1

Algorítmica

Introducción

Ordenación

Burbuja

Selección

Inserción

Ordenación  
rápida

Quick Sort

**Heap Sort**

Merge Sort

Floyd

Hanoi

La gráfica empírica obtenida ha sido:

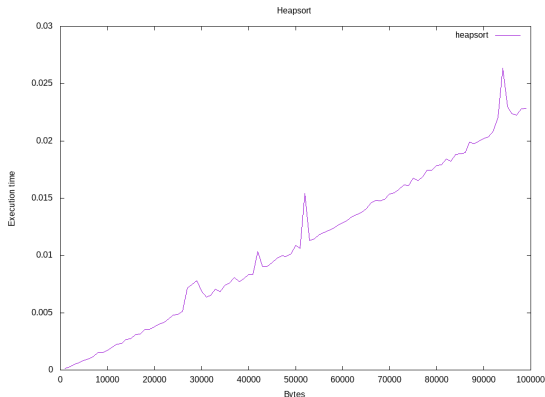


Figura: Heapsort, gráfica empírica.



# Heap Sort

Práctica 1

Algorítmica

Introducción

Ordenación

Burbuja

Selección

Inserción

Ordenación  
rápida

Quick Sort

**Heap Sort**

Merge Sort

Floyd

Hanoi

La gráfica híbrida obtenida ha sido:

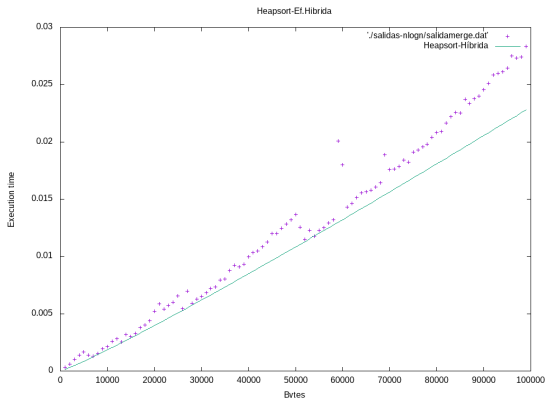


Figura: Heapsort, gráfica híbrida.

# Heap Sort

## Práctica 1

### Algorítmica

#### Introducción

#### Ordenación

##### Burbuja

##### Selección

##### Inserción

#### Ordenación rápida

##### Quick Sort

##### Heap Sort

##### Merge Sort

#### Floyd

#### Hanoi

Este algoritmo lo hemos ejecutado en otra máquina más potente, de esta manera, podemos ver como varían los tiempos de ejecución del mismo algoritmo, ejecutado en dos máquinas distintas.

- Procesador (frecuencia): Intel Core i7-5500U (3.0 GHz x 2)
- Memoria RAM: 4 GB
- Disco duro: SSD 256 GB
- S.O: Linux Mint 17.3 Cinnamon 64-bit

# Heap Sort

Práctica 1

Algorítmica

Introducción

Ordenación

Burbuja

Selección

Inserción

Ordenación  
rápida

Quick Sort

**Heap Sort**

Merge Sort

Floyd

Hanoi

El ajuste de ambas gráficas es el siguiente:

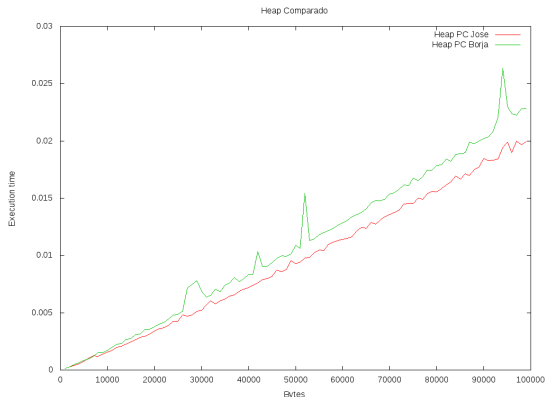


Figura: Comparación en dos máquinas. Heapsort.

# Merge Sort

Práctica 1

Algorítmica

Introducción

Ordenación

Burbuja

Selección

Inserción

Ordenación  
rápida

Quick Sort

Heap Sort

Merge Sort

Floyd

Hanoi

La gráfica empírica obtenida ha sido:

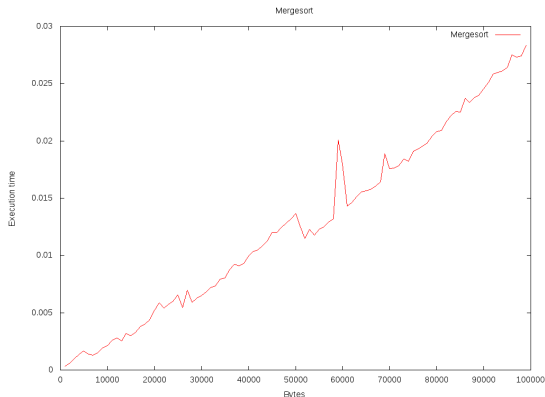


Figura: Mergesort, gráfica empírica.

# Merge Sort

La gráfica híbrida obtenida ha sido:

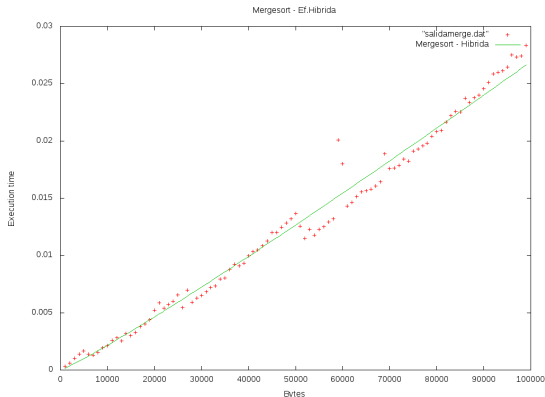


Figura: Mergesort, gráfica híbrida.

# Algoritmos $n * \log(n)$

## Práctica 1

### Algorítmica

#### Introducción

#### Ordenación

##### Burbuja

##### Selección

##### Inserción

#### Ordenación

##### rápida

##### Quick Sort

##### Heap Sort

##### Merge Sort

#### Floyd

#### Hanoi

Por último, mostramos el porcentaje de error así como las constantes ocultas.

Algoritmo	Constante Oculta	Error
Mergesort	$a_0 = 2.33821e-08$	$\pm 1.564e-10$ (0.6689 %)
Quicksort	$a_0 = 1.43368e-08$	$\pm 7.621e-11$ (0.5315 %)
Heapsort	$a_0 = 2.00227e-08$	$\pm 1.222e-10$ (0.6104 %)

# Floyd

## Práctica 1

### Algorítmica

#### Introducción

#### Ordenación

Burbuja

Selección

Inserción

#### Ordenación rápida

Quick Sort

Heap Sort

Merge Sort

#### Floyd

#### Hanoi

El algoritmo de Floyd, a diferencia de los anteriores, no es un algoritmo de ordenación. Su función, es la de encontrar el camino mínimo en grafos. La eficiencia teórica de este algoritmo es  $O(n^3)$ . Además de analizar la eficiencia empírica e híbrida, hemos realizado un ajuste erróneo, para demostrar que efectivamente la eficiencia de este algoritmo es la anteriormente mencionada.

# Floyd

Práctica 1

Algorítmica

Introducción

Ordenación

Burbuja

Selección

Inserción

Ordenación  
rápida

Quick Sort

Heap Sort

Merge Sort

**Floyd**

Hanoi

La gráfica empírica obtenida ha sido:

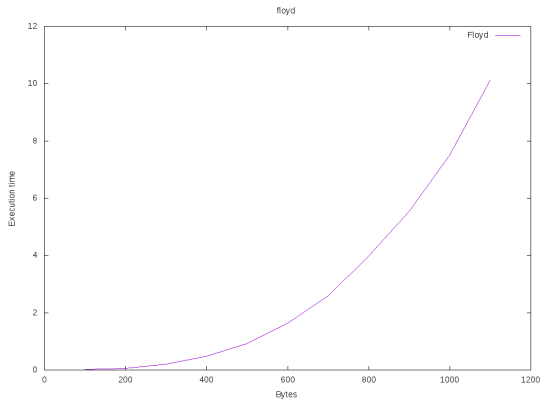


Figura: Floyd, gráfica empírica.



# Floyd

Práctica 1

Algorítmica

Introducción

Ordenación

Burbuja

Selección

Inserción

Ordenación  
rápida

Quick Sort

Heap Sort

Merge Sort

Floyd

Hanoi

La gráfica híbrida obtenida ha sido:

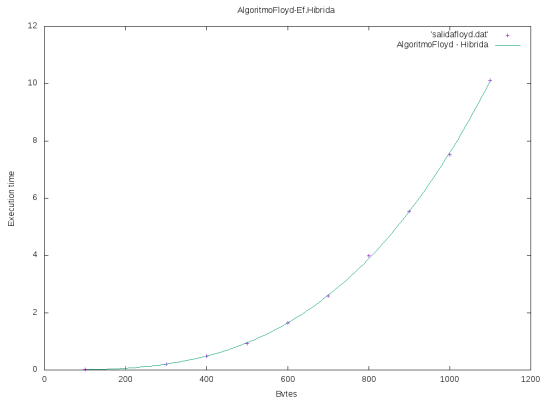


Figura: Floyd, gráfica híbrida.

# Floyd

## Práctica 1

### Algorítmica

#### Introducción

#### Ordenación

Burbuja

Selección

Insertión

#### Ordenación rápida

Quick Sort

Heap Sort

Merge Sort

#### Floyd

#### Hanoi

Ajuste híbrido erróneo ( $O(n^3)$  ajustada a una función  $O(n * \log(n))$ ).

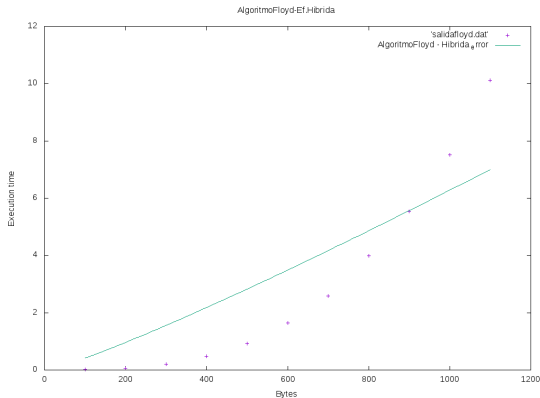


Figura: Floyd, ajuste erróneo.

# Floyd $O(n^3)$

## Práctica 1

### Algorítmica

#### Introducción

#### Ordenación

Burbuja

Selección

Inserción

#### Ordenación rápida

Quick Sort

Heap Sort

Merge Sort

#### Floyd

#### Hanoi

Por último, mostramos el porcentaje de error así como las constantes ocultas de ambos ajustes.

Algoritmo	Constante Oculta	Error
Floyd	$a_0 = 7.59068e-09$	$\pm 2.163e-11$ (0.2849 %)
Floyd erróneo	$a_0 = 0.000908818$	$\pm 0.0001093$ (12.03 %)

# Hanoi

## Práctica 1

### Algorítmica

#### Introducción

#### Ordenación

Burbuja

Selección

Inserción

#### Ordenación rápida

Quick Sort

Heap Sort

Merge Sort

#### Floyd

#### Hanoi

El algoritmo de Hanoi, se encarga específicamente de resolver el problema de las torres de Hanoi. Este algoritmo presenta una eficiencia teórica de  $O(2^n)$ , lo que hace que el número de entradas para este algoritmo sea muy reducido.

# Hanoi

## Práctica 1

### Algorítmica

#### Introducción

#### Ordenación

##### Burbuja

##### Selección

##### Inserción

#### Ordenación rápida

##### Quick Sort

##### Heap Sort

##### Merge Sort

#### Floyd

#### Hanoi

La gráfica empírica obtenida ha sido:

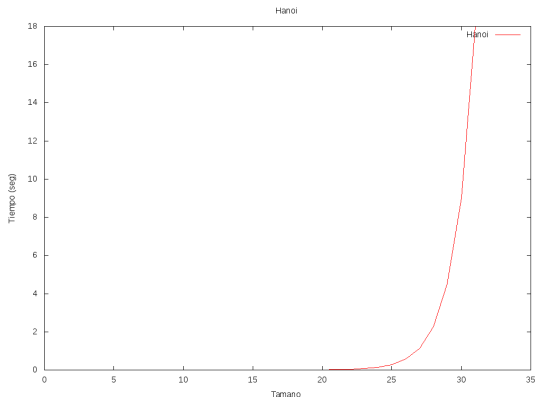


Figura: Hanoi, gráfica empírica.



# Hanoi

## Práctica 1

### Algorítmica

#### Introducción

#### Ordenación

Burbuja

Selección

Inserción

#### Ordenación rápida

Quick Sort

Heap Sort

Merge Sort

#### Floyd

#### Hanoi

En este algoritmo hemos hecho varias comparaciones:

- Usando diferentes optimizaciones a la hora de compilar.
- Utilizando un lenguaje de programación distinto (Python vs C++).







# Hanoi $O(2^n)$

## Práctica 1

### Algorítmica

#### Introducción

#### Ordenación

Burbuja

Selección

Inserción

#### Ordenación rápida

Quick Sort

Heap Sort

Merge Sort

#### Floyd

#### Hanoi

Por último, mostramos el porcentaje de error así como las constantes ocultas.

Algoritmo	Constante Oculta	Error
Hanoi	$a_0 = 8.38461e-09$	$\pm 3.095e-12$ (0.03691 %)

## Práctica 1

### Algorítmica

#### Introducción

#### Ordenación

Burbuja

Selección

Inserción

#### Ordenación rápida

Quick Sort

Heap Sort

Merge Sort

#### Floyd

#### Hanoi

# ¿Preguntas?