Practica 1

Una practica escrita en beamer

2º Doble Grado Infomática y Matemáticas

Índice

En estas diapositivas desarrollaremos como hemos realizado la prática, con el objetivo de que quede claro como se resuelven y el por qué de sus resultados.

Enunciado

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 l amemoria tablas diferentes paara los algoritmos de distinto orden de eficiencia (una con algoritmos $O(n^2)$, otra con los $O(nlog(n))$, otra con los $O(n^3)$ y otra con los $O(2^n)$.

Para facilitar los numerosos casos de prueba que suceden, realizamos un script en bash para automatizar los procesos de compilación y ejecución de los scripts. Utilizamos el lenguaje bash porque consideramos que facilita el uso de programa y scripts externos.

Código del Script

AQUÍ VA EL CÓDIGO DEL SCRIPT, HELP PLEASE

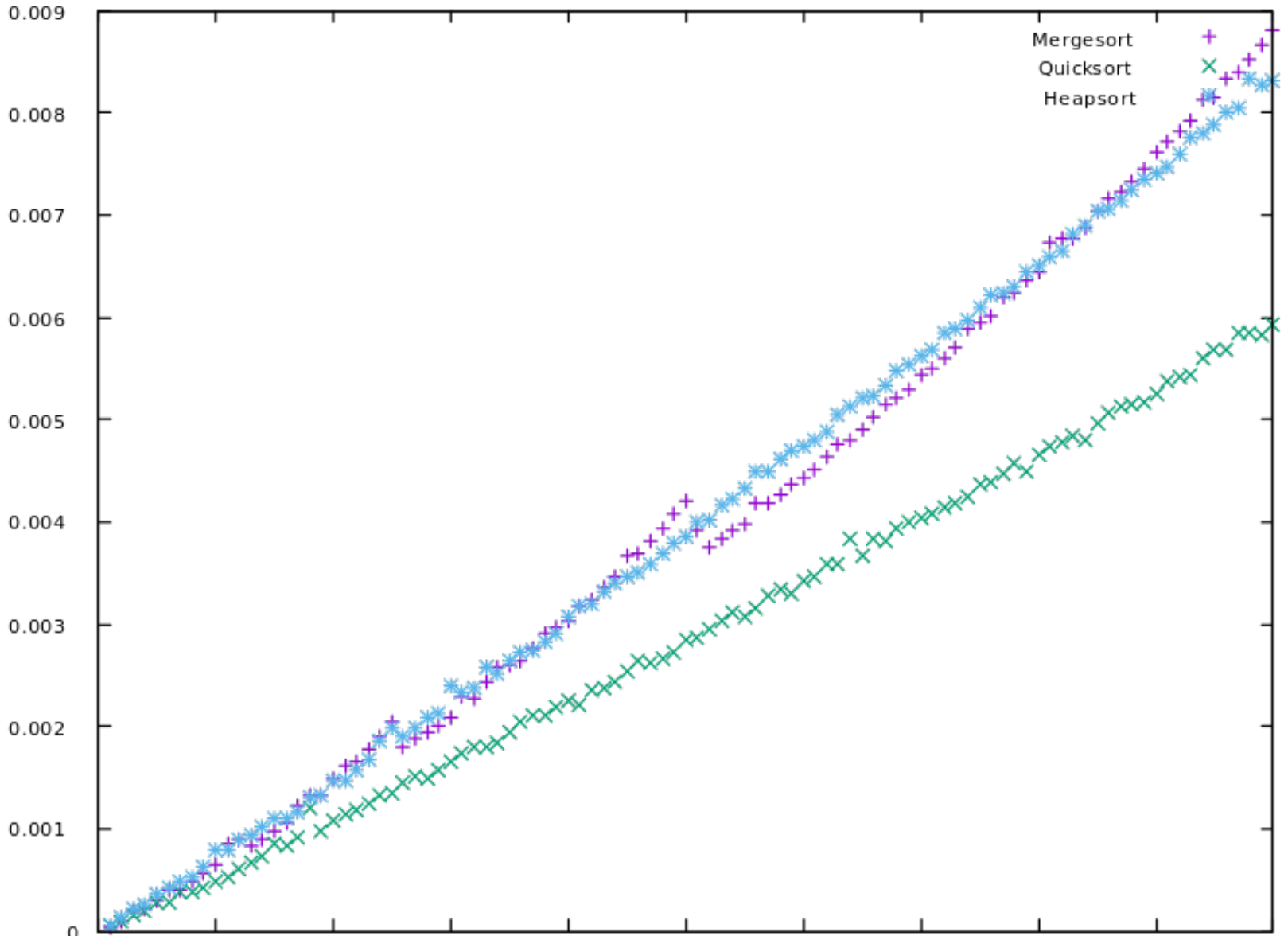
Como podemos observar en el código, primero buscamos todos los archivos .cpp, para proseguir con su compilado. Después generamos o vaciamos el archivo ../resultado/ejecutable.dat. Por último ejecutamos un bucle while de estilo for, con el número máximo de la variable y cuanto aumenta en el paso a paso.

A cada orden de comlejidad le hemos definido una cantidad diferente de elementos con los que trabajar.

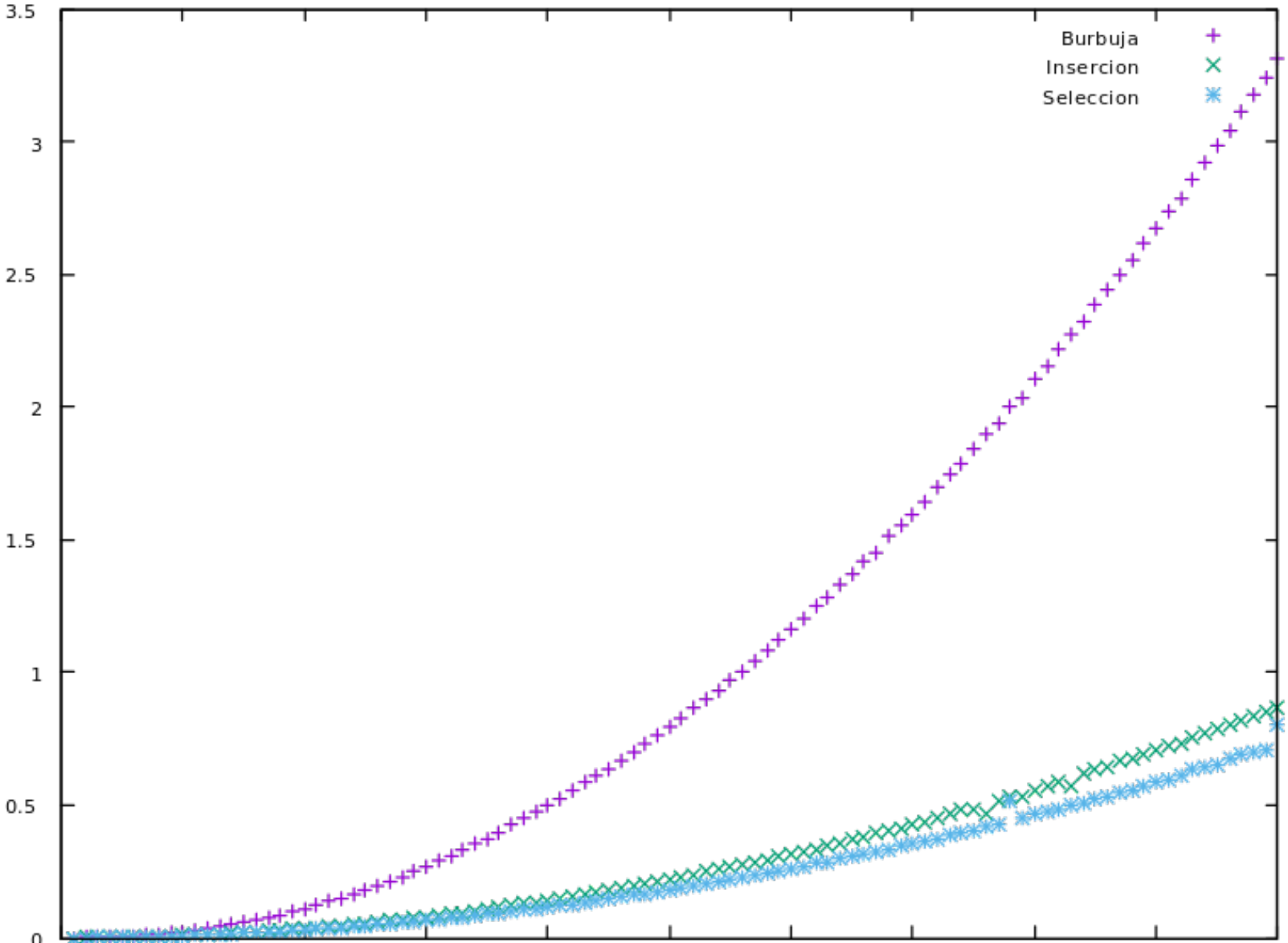
- Para los algoritmos $nlog(n)$ (mergesort, quicksort, heapsort) hemos definido que empieze en 1000, y llegue a 100000 subiendo de 1000 en 1000. Un total de 100 iteraciones.
- Para los algoritmos n^2 (burbuja, insercción, selección) hemos definido que empieze en 500, y suba hasta 500000. Un total de 100 iteraciones.
- El algoritmo Floyd de orden n^3 parte de 25, y sube de 25 en 25 hasta 2500. Un total de 100 iteraciones.
- El algoritmo hanoi de orden 2^n parte de 1 y va subiendo unidad a unidad hasta 28. Un total de 28 iteraciones.

Con el objetivo de poner de manifiesto de un modo más accesible la información, exponemos la gráfica, necesarias para el ejercicio siguiente.

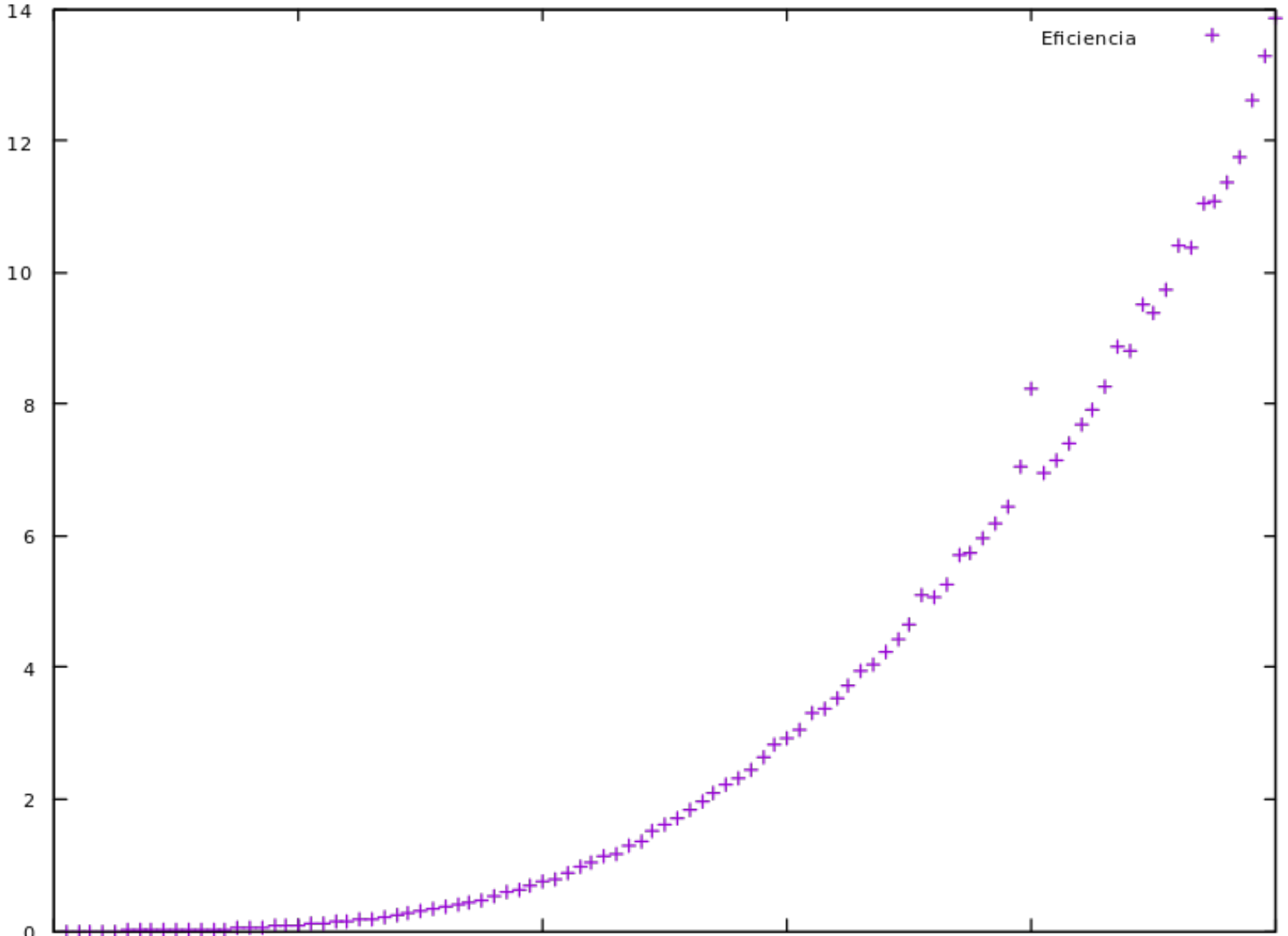
tablas orden $nlog(n)$



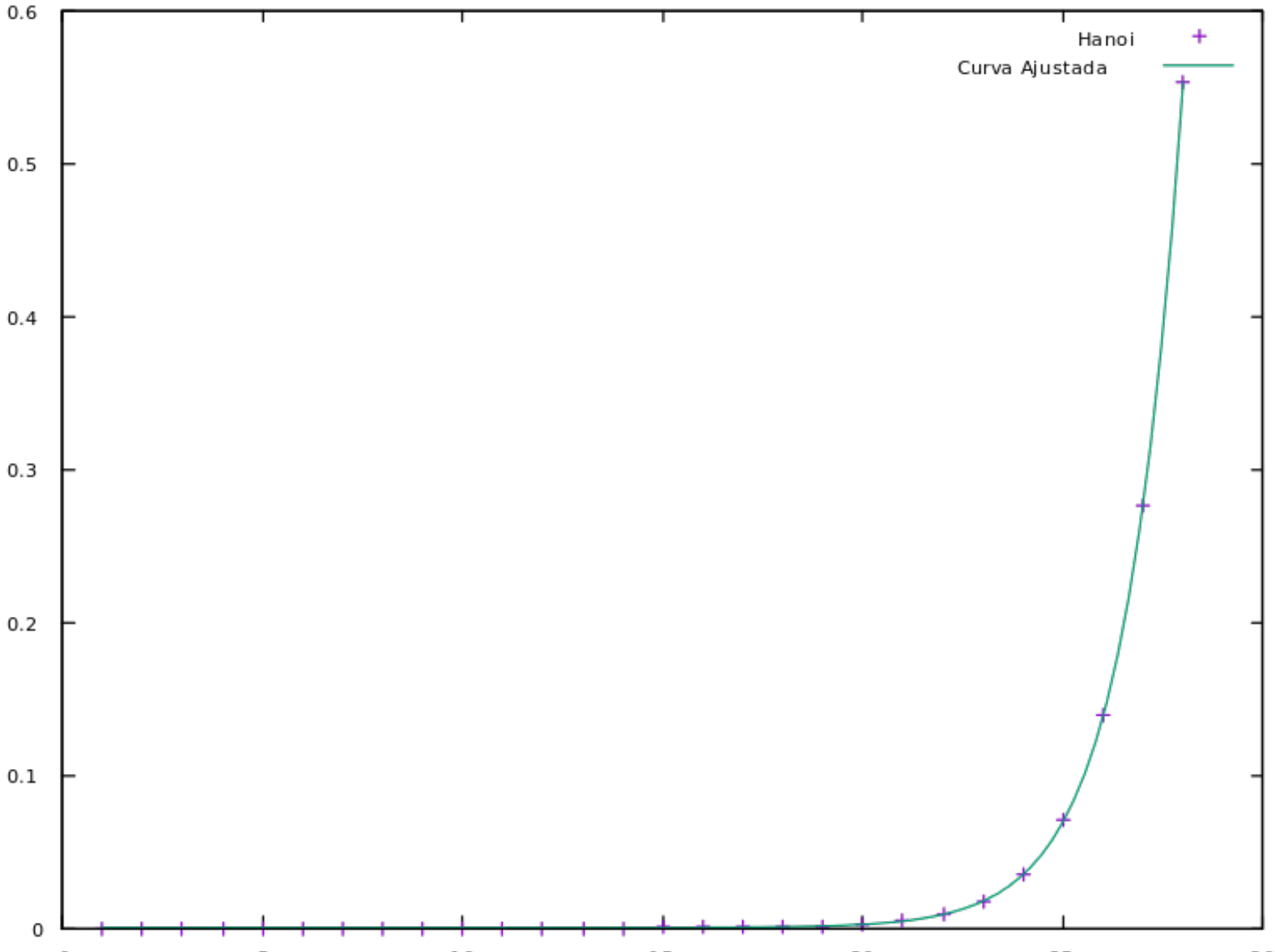
tablas orden n^2



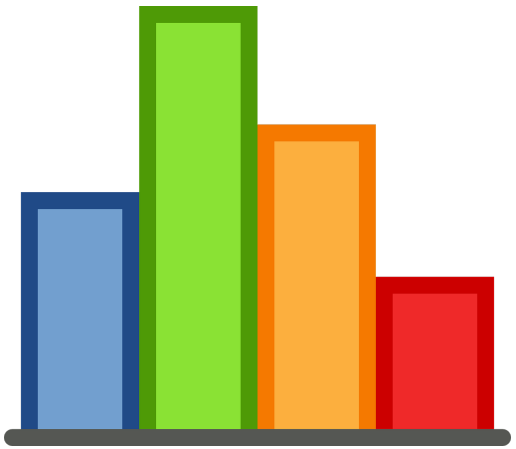
tablas orden n^3



tablas orden 2^n



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 (ordenar) exponga una tabla compartida dondea poder apreciar las diferencias en rendimiento de algoritmos con diferente orden de eficiencia.



Las gráficas por ordenes de eficiencia ya las mostramos en el ejercicio anterior. Ahora proseguiremos explicando como las generamos y exponiedo como las hallamos (uso de gnuplot)

Comparación algoritmos ordenación