

Practica 1

Analisis de eficiencia

ADRIAN ORDUÑA DÍAZ, RAFAEL LEYVA RUIZ
Universidad de Granada
16 de octubre de 2016

1. Introduccion

En esta primera práctica se ha analizado la eficiencia de diversos algoritmos tanto empíricamente como teóricamente, además se ha procedido a comprobar como se comportan los algoritmos en diversos casos, tanto extremos (mejor y peor caso) como evaluando como afecta el entorno, el proceso de compilación, etc. a las implementaciones y eficiencia de estos en el mundo real. Los algoritmos evaluados han sido los siguientes:

- Búsqueda lineal.
- Ordenación lineal o algoritmo burbuja.
- Búsqueda binaria.
- Multiplicación de matrices.
- Ordenación por mezcla o mergesort.

Nota: Los ejercicios descritos en este documento no mantienen el mismo orden que en la presentación de la práctica, ya que cuando se comience a hablar de un algoritmo se detallarán todos los análisis y pruebas realizadas, no obstante antes de cada análisis se detallará el ejercicio en concreto del que se trata.

2. Búsqueda lineal

El primer algoritmo analizado fue el algoritmo básico de búsqueda, el primero en el que pensaría cualquiera cuando se le plantea el problema, que no es más que recorrer el vector hasta encontrar el elemento buscado. De esta descripción se aprecia claramente que en su peor caso es lineal, ya que habría que recorrer todo el vector, si el elemento buscado aparece en la última posición.

2.1. Analisis de eficiencia teorico

El desarrollo matemático de el analisis de la eficiencia teorica del codigo del algoritmo que se muestra arriba es el siguiente:

Tras hacer pruebas con distintos tamaños de entradas se ha obtenido la siguiente grafica de tiempos, en la cual se enfrenta el tamaño del vector en el que se ha buscado aplicando el algoritmo (eje x) y el tiempo que ha tardado en completarse el algoritmo para cada tamaño.

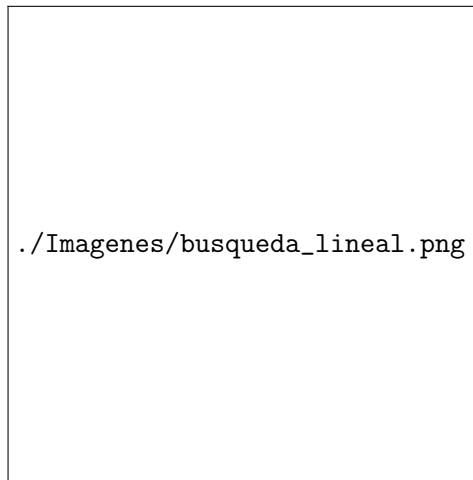


Figura 1: Datos de la salida del algoritmo de busqueda lineal

Como se aprecia el crecimiento de los tiempos de ejecucion del algoritmo es lineal conforme crece el tamaño de las entradas del problema. Tras obtener los datos anteriores, se uso la funcion fit de gnuplot para ajustar dicha salida con una funcion, para ver como se corresponde el tiempo teórico y el empirico, para ello se ajusto a la funcion $f(x) = a * x$, de tal modo que a nos indica la constante oculta del algoritmo. La grafica resultante es la siguiente:

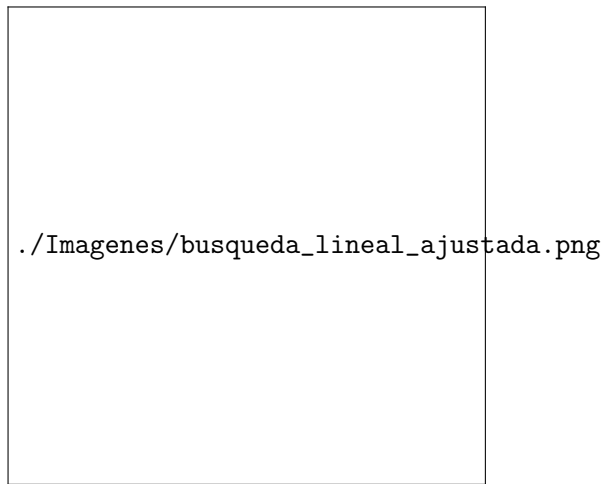


Figura 2: Salida del algoritmo de busqueda lineal ajustada a $f(x)$