

Medición de tempos de algoritmos de ordenación

Alejandro José López Barcia

24 de marzo de 2018

Índice

1. Introducción	3
2. Algoritmo de quicksort	4
3. Algoritmo de burbulla	5
4. Algoritmo de selección	6
5. Conclusiones	7

1. Introducción

Dado un vector de n elementos xerados automaticamente, comprobarase, empregando tres algoritmos diferentes, o tempo que tarda en ordenarse o vector, que empezará cun tamaño dado, incrementarse ao acabar a ordenación e chegará a un tamaño límite indicado ao executar o programa.

Tanto o tamaño inicial, como o número de elementos que se incrementa e o tamaño límite do vector indicaranse nas respectivas seccións dos algoritmos a tratar. Os algoritmos a empregar son:

- Algoritmo de quicksort:
 - Orde superior: $\mathcal{O}(n^2)$ (cuadrático)
 - Orde inferior: $\Omega(n \log n)$ (superlineal)
 - Orde promedia: $\Theta(n \log n)$ (superlineal)
- Algoritmo de burbulla:
 - Orde superior: $\mathcal{O}(n^2)$ (cuadrático)
 - Orde inferior: $\Omega(1)$ (constante)
 - Orde promedia: $\Theta(n^2)$ (cuadrático)
- Algoritmo de selección:
 - Orde superior: $\mathcal{O}(n^2)$ (cuadrático)
 - Orde inferior: $\Omega(n)$ (lineal)
 - Orde promedia: $\Theta(n^2)$ (cuadrático)

O equipo no que se realiza a experimentación dos algoritmos dispón dunha CPU Intel i7-3770k @ 3.5 GHz.

2. Algoritmo de quicksort

Pártese dun tamaño inicial de 10.000 elementos, incrementando o vector en 10.000 e establécese como tamaño final un vector de 5.000.000 elementos.

Como se pode apreciar na figura 1, a gráfica presenta un aspecto superlineal, na cal a ordenación de ata 60.000 elementos mantense arredor dos 0,016 segundos (dende 10.000 ata 60.000 obtéñense os mesmos resultados) e para ordenar 5.000.000 elementos tárdase 2,531 segundos.

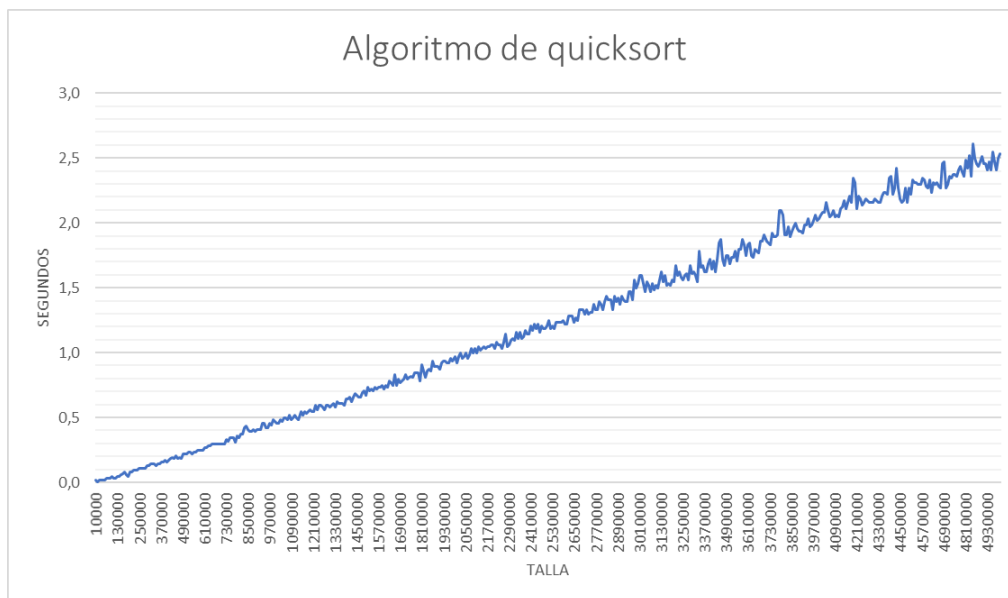


Figura 1: Algoritmo de quicksort

3. Algoritmo de burbulla

Pártese dun tamaño inicial de 10.000 elementos, incrementando o vector en 10.000 e establécese como tamaño final un vector de 500.000 elementos, ao presentar unha orde promedia de $\Theta(n^2)$.

Como se pode observar na figura 2, a gráfica presenta un aspecto cuadrático, na cal a ordenación inicial de 10.000 elementos tarda 0,421 segundos e a seguinte ordenación de 20.000 elementos tarda 1,672 segundos, ate un máximo de 1.091,563 segundos para 500.000 elementos.

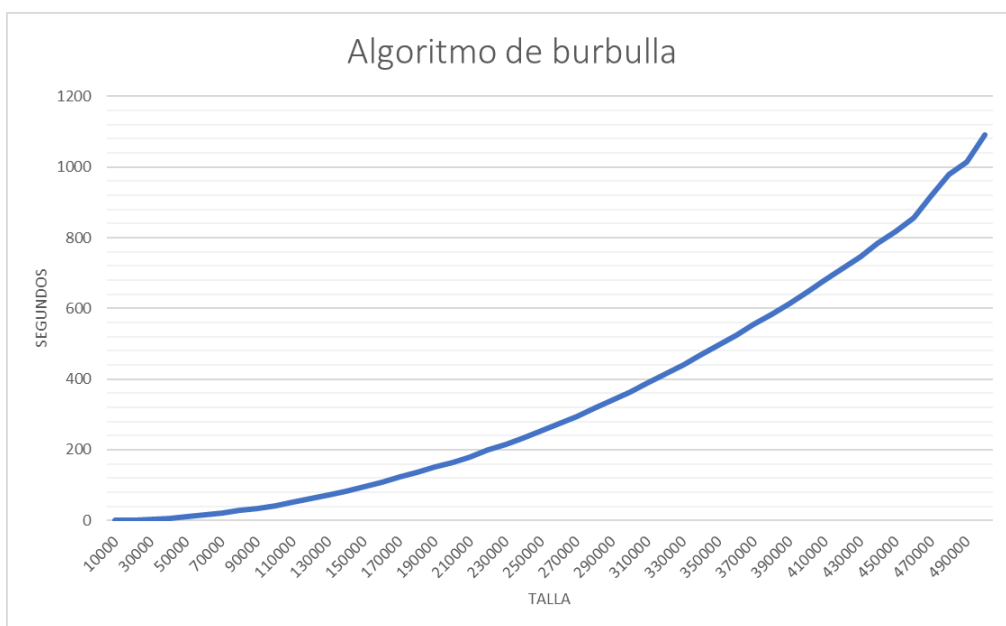


Figura 2: Algoritmo de burbulla

4. Algoritmo de selección

Pártese dun tamaño inicial de 10.000 elementos, incrementando o vector en 10.000 e establécese como tamaño final un vector de 500.000 elementos, polos mesmos motivos que no apartado anterior.

Como se pode contemplar na figura 3, a gráfica presenta un aspecto cuadrático, máis pronunciado, na cal para a ordenación inicial de 10.000 elementos tarda 0,531 segundos e a seguinte ordenación de 20.000 elementos acádase en 2,14 segundos, ate un máximo de 1.287,109 segundos para o vector de 500.000 elementos.

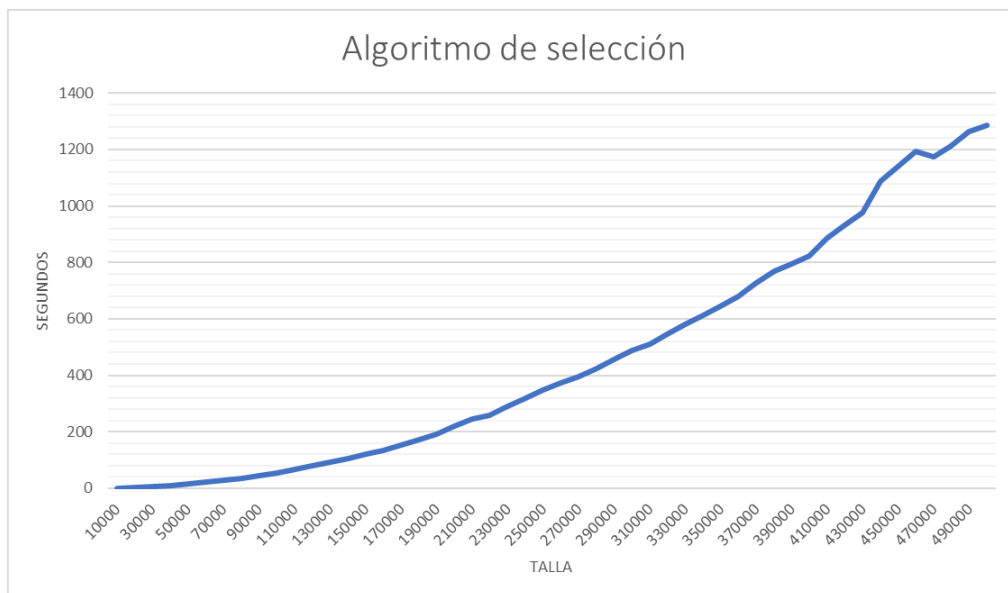


Figura 3: Algoritmo de selección

5. Conclusións

Considerando como tempo razoable un intervalo de 10 segundos, o algoritmo de burbulla tardou 10,453 segundos manexando 50.000 elementos e o algoritmo de selección tardou 8,562 segundos para un tamaño de 40.000 elementos, mentras que o algoritmo de quicksort pode manexar vectores de 5 millóns de elementos en menos de 3 segundos (non se chegou a probar unha talla o suficientemente grande como para tardar 10 segundos).

En xeral, e especialmente para tallas grandes, o algoritmo de quicksort é moito máis rápido que os outros dous, podendo facer unha ordenación **dez veces maior en menos de 3 segundos**, mentras que tanto o de burbulla e selección tardan máis de 1.000 segundos para unha talla dez veces menor.

Tomando como tamaño de vector, o tamaño inicial de 10.000 elementos ou menos, poderíanse empregar indistintamente calquera dos tres algoritmos, posto que aínda que o quicksort tarde 0,016 segundos e os outros dous 0,421 e 0,531, ámbolos tres tardan menos dun segundo, tempo que se consideraría inapreciable.