

PROTOCOL: MergeSort és més ràpid que QuickSort?

Roser González Valeri, José Antonio Rius Cobo i Erik Carrasco Alastrué.

Novembre 2021

Introducció i objectius

En l'àmbit de la informàtica, el disseny de programes té una importància cabdal. Aquests han de complir unes determinades característiques: llegibilitat, robustesa, reusabilitat, eficiència etc...

Volem veure si podem fer més ràpida certa aplicació. Ara hi ha implementat l'algoritme Quicksort (Q) d'ordenació de vectors, però volem decidir si convé canviar-ho per Mergesort (M) –en cas de que tingui major eficiència temporal. Per això, intentarem rebutjar la hipòtesi nul·la de igualtat de temps a favor de la alternativa unilateral de que M és més ràpid que Q.

Variables recollides

La nostra variable d'interès Y serà el temps d'execució de l'algorisme mesurat en segons. La nostra variable X serà el tipus d'algorisme emprat per endreçar el vector (*MergeSort* o *QuickSort*). A més, recollirem una variable Z que serà la longitud (número d'elements) del vector.

Recollida de dades

Hem generat al atzar 50 vectors de grandària entre 0 i 10^6 amb números d'una Distribució Uniforme. Els ordenarem amb els dos algorismes, alternant aleatòriament l'ordre d'execució.

Emprarem el programari R v.2.11.1 amb un Intel Core 2 de 2.66GHz i 3.46 Gb de memòria RAM; i amb Microsoft Windows XP.

Recollirem la grandària del vector i el temps d'execució d'ambdós algorismes. Com el temps d'execució depèn de la grandària [$O(n \log(n))$] estudiarem si és més simple resumir els resultats amb una diferència o un rati –que hem obtingut a partir de la diferència dels logaritmes naturals de les variables.

Les implementacions de cadascun dels algorismes s'han extret dels següents enllaços:

QuickSort: <https://en.wikipedia.org/wiki/Quicksort>

MergeSort: https://en.wikipedia.org/wiki/Merge_sort

Mostra

Per aconseguir una mostra aleatòria simple de la població de vectors a ser ordenats, hem decidit tenir diferents mides de vectors i generar-los de forma aleatòria. D'aquesta forma, tant la mida com els propis elements del vector estan generats de forma aleatòria proporcionant una mostra de 50 vectors de diferents mides. El nostre disseny és amb dades aparellades ja que cadascun dels vectors l'endregarem amb els 2 algorismes.