

EXPLICACIÓ DE LES ESTRUCTURES DE DADES I ALGORISMES

La característica principal del nostre programa és l'important ús que li donem a una estructura de dades que ens proporciona Java: les LinkedList.

La LinkedList ens permet fer interseccions o eliminacions en temps constant utilitzant iteradors, encara que l'accés als elements és seqüencial.

El principal benefici del qual nosaltres ens aprofitem és la reutilització d'iteradors existents per insertar, eliminar o modificar cel·les (amb contingut o sense).

Un altre benefici, encara que menys habitual, és quan es voldria afegir o eliminar la primera posició de la llista (o de les primeres), ja que aquesta operació amb LinkedList triga un temps $O(1)$. Amb una altra estructura de dades com podria ser ArrayList, el fet d'haver de moure totes les posicions un lloc faria que el temps fos $O(n)$, provocant una notable diferencia de temps comparat amb la estructura escollida per nosaltres.

Un "inconvenient" de les LinkedList podria ser que si hem d'emmagatzemar una fulla molt gran (amb continguts), els punters a l'element següent i l'anterior s'han de guardar, provocant una major sobrecarga, però si aquesta fulla que ha de ser molt gran, encara no hi té continguts, no ha de reservar un espai de memòria que si hauria de reservar si s'utilitza una altra estructura com les ArrayList, les quals reserven tant espai en memòria com necessiten abans de res.

Algorismes a comentar:

La cerca d'un element en un rang seleccionat té cost $O(n)$ ja que fem la cerca lineal, degut a que el rang no té perquè estar ordenat i pot ser que contingui cel·les que no siguin del tipus demanat(String).

El reemplaçament d'un element en un rang seleccionat també té cost $O(n)$ ja que es basa en la cerca d'un element, ja explicat abans, i després en la modificació d'aquest contingut, el qual és constant.

L'algorisme que utilitzem per ordenar un rang de cel·les és el Merge Sort (tant ascendent com descendent). El cost d'aquesta operació és $O(n^2 \log(n))$.

Les operacions que tenen a veure amb un rang (Average, Median, Variance && StandradDeviation) o amb dos rangs (Covariance && Pearson) tenen cost $O(n)$ ja que accedim linealment als continguts de les cel·les per després poder fer les operacions que tenen costs constant.

Copiar i enganxar un rang en una altra posició també té cost $O(n)$, per la mateixa raó de l'accés lineal.