

Laboratorio N° 1
Estructuras de Datos y Algoritmos

Fecha: Jueves 01 de septiembre de 2021

Enunciado

La recursión es una forma de modelamiento de algoritmos que consiste en usar llamadas a una misma función (método en nuestro caso) desde su propio código para resolver casos distintos y/o “contenidos” en el caso original. Posteriormente, al resolver el caso básico, se utiliza el resultado más básico en el cálculo de casos más complejos hasta completar la resolución del caso inicialmente requerido.

Asimismo, las colecciones son objetos que agrupan múltiples instancias de otros objetos y/o elementos. Son fundamentales a la hora de trabajar con valores múltiples y agrupar e organizar datos para su procesamiento. En particular, se pueden usar colecciones para implementar estructuras de datos básicas como son listas, pilas, colas y mapas.

Finalmente, los algoritmos se evalúan en términos de su complejidad de modo de discernir si un algoritmo es mejor que otro en cuanto al tiempo de ejecución.

El foco de este laboratorio va a ser de resolver problemas simples utilizando colecciones y estructuras de datos simples de forma predominantemente recursiva. Asimismo, se evaluará la complejidad temporal de dichos algoritmos de modo de ayudar a entrar en confianza con el diseño de algoritmos recursivos, así como de su evaluación.

Específicamente:

1. **(0.5 pts)** Genere un archivo llamado números.txt que albergue K números enteros aleatorios (K es un parámetro de entrada) entre 0 y 1000. El proceso de generación se debe efectuar recursivamente.
2. **(1.5 pts)** Almacene 1000000 de valores enteros aleatorios en un ArrayList (léidos desde números.txt) y cuente recursivamente cuántos valores mayores, menores e iguales a V (parámetro de entrada) hay en el arreglo. (Hint: se puede inspirar en la búsqueda binaria). Cuente además cuantas comparaciones efectúa para llegar al resultado final.
3. **(1 pto)** Ordene los valores en el ArrayList del punto 2. Cuente cuantas comparaciones efectúa para lograr un ArrayList ordenado.

4. **(2 ptos)** Repita la tarea del punto 2. usando el ArrayList ordenado. Implemente además un método que cuente recursivamente cuantos valores mayores, menores e iguales a V (parámetro de entrada) hay en el arreglo que use menos comparaciones que en el punto 2.
5. **(1 pto)** Guarde en un HashMap los conteos asociados a cada valor entre 0 y 1000 (donde dichos valores serían K, la llave y sus respectivos conteos V, los valores)

Condiciones de entrega

Se indican las siguientes condiciones para la entrega de la tarea:

- La tarea se desarrolla **en parejas**. Los códigos serán sometidos a comprobación automática de plagio.
- Los códigos fuentes deben estar escritos en Java. Es requisito indispensable que los códigos compilen y ejecuten **sin cambios**. De no cumplirse esta condición, el código correspondiente será evaluado con nota mínima de inmediato. Se le exige que únicamente utilice librerías estándar.
- El informe estará en formato PDF. (Si hay un informe que entregar) **
- Deberá indicar al profesor la conformación de los grupos a más tardar el día 01 de septiembre de 2021 por medio del foro de Canvas, whatsapp, etc...
- Todos los archivos deben colocarse en un único archivo .zip con nombre <GrupoX>.zip

La tarea se entregará vía el Canvas de la sección. La fecha y hora límite para la entrega son el día Jueves 22 de Septiembre de 2021 a las 23:59. Cualquier entrega atrasada no se considerará y será calificada con la nota mínima de inmediato.