

Proyecto #1: Métodos de ordenamiento externo

Estructura de datos y algoritmos II

Cabello, Sofía López, Ricardo Núñez, Diego

13 de noviembre de 2020

1. Polifase

Polifase es un algoritmo de ordenamiento externo que requiere de un ordenamiento interno y de 3 archivos auxiliares. El algoritmo básicamente consiste separar la colección inicial de datos en pequeños grupos ordenados, y después, a través de merge, unirlos en uno solo.

Es necesario un algoritmo de ordenamiento interno para ordenar los pequeños grupos de de datos. Los grupos de datos deben caber en memoria interna por esta razón, así que el tamaño de los grupos se suele elegir en función del tamaño de la memoria interna.

El proceso de generación de estos bloques es simple. Primero, se lee el grupo de datos del archivo de entrada(F0) y se ordenan los datos con un algoritmo de ordenamiento interno. Para este proyecto se eligió Insertion Sort, porque el tamaño de los bloques se fijó en 10 datos y porque es un algoritmo sencillo y eficiente para cantidades de datos pequeñas. Después, el grupo ordenado se escribe en un archivo auxiliar (F1) y se repite el proceso para el siguiente grupo, solo que en esta ocasión se escribe en otro archivo auxiliar (F2).

Después de tener los pequeños grupos ordenados, se hace merge entre pares de ellos para generar uno más grande. Este proceso se repite hasta que quede solo un grupo con todos los datos ordenados.

Para hacer el merge, los archivos F1 y F2 se convierten en los archivos de lectura. Luego, el primer bloque de F1 se intercala con el primer bloque de F2 y se escribe el bloque resultante en F0. Luego, se repite el proceso con el segundo bloque de cada archivo, pero esta vez el resultado se escribe en F3. Cuando se terminan los bloques de F1 y F2, F0 y F3 se vuelven los archivos de lectura y F1 y F2 en los de salida. Se repite el proceso hasta que todos los datos estén en un solo archivo.

2. Clase File

Todos los sistemas operativos requieren de pathnames para nombrar archivos y directorios. Esta clase es una vista abstracta e independiente del sistema operativo de los pathnames. Los pathnames consisten en un prefijo específico del sistema y una secuencia de cero o más directorios separados por un carácter también determinado por el sistema(En sistemas unix-like “/” y en Windows “\”). Además, los pathnames pueden ser absolutos o relativos. Los absolutos son las rutas completas desde la raíz del sistema y los relativos son solo parte de ellas. File por defecto resuelve las rutas relativas al directorio en el que se invocó a la JVM.

Las instancias de esta clase no necesariamente refieren a archivos o directorios que realmente existen. Si representan a un archivo o directorio real, entonces ese archivo o directorio debe existir en el sistema de archivos. Además, las instancia de esta clase no son mutables, por lo que no pueden ser modificadas después de su creación.

El sistema de archivos puede administrar los permisos de archivos y directorios de los cuales esta clase es dependiente. Estos permisos pueden hacer que algunos métodos de esta clase fallen.

Algunos de los métodos que utilizamos de esta clase en el proyecto son:

- `createNewFile()`: Crea un nuevo archivo si y solo si el archivo aun no existe.
- `delete()`: Elimina el archivo si es que existe. Si la instancia refiere a un directorio entonces este debe estar vacío.
- `exists()`: Verifica que el archivo denotado por el pathname exista.
- `getAbsolutePath()`: regresa una cadena con la ruta absoluta del archivo.
- `isDirectory()`: Verifica si la instancia refiere a un directorio.
- `listFiles()`: Devuelve un arreglo de objetos File de todos los archivos dentro de un directorio.
- `mkdir()`: Crea un directorio en la ruta de la instancia.

3. Analisis de Polifase

La implementación de polifase se divide en cuatro clases agrupadas en el paquete polifase, y utiliza a las clases dato y alumno del paquete dato. Los

nombres de las clases son Polifase, Merge, InsertionSort y FilesDirect.

3.1. Clase Polifase

La clase polifase es el método principal de polifase. Dentro de ella se pueden encontrar tres variantes del método sort, una para cada criterio de ordenamiento. Lo mismo sucede con las clases Merge e InsertionSort. Para este análisis y explicación del programa, se tomara en cuenta solo la versión para número de cuenta y se mencionarán las diferencias con las variantes.

El método sortNum recibe como parámetro el nombre del archivo que se va a ordenar. Algo importante que notar es que recibe únicamente el nombre del archivo y no su ruta completa, ya que esta se genera más adelante.

Lo primero que hace este método es instanciar algunas de las clases necesarias, como FilesDirect, File, FileReader, BufferedReader, Merge y Dato. Con el método rutaFolder de FilesDirect se obtiene la ruta del folder donde se