Universidad Nacional Autónoma de México

Facultad de Ingeniería

División de Ingeniería Eléctrica Electrónica



**Proyecto 1**

Implementación de algoritmos de ordenamiento externo: Polifase, Mezcla Directa y Radix

**Integrantes**

Monsalvo Bolaños Melissa Monserrat

Cárdenas Cárdenas Jorge

Garrido Sánchez Samuel Arturo

Perteneciente a la asignatura de

Estructuras de datos y algoritmos II

Asignatura impartida por

M.I. Tista García Egdar

Grupo 5

Semestre 2019 - 1

Ciudad Universitaria, Coyoacán, Ciudad de México



**Introducción**

El ordenamiento de sistemas son útiles desde la antiguedad dado que nuestro razonamiento lógico radica en encontrar y utilizar dichos objetos del sistema para nuestro fin, sin embargo para que esto suceda de la mejor manera es necesario colocar el objeto en el lugar que le corresponda.

El razonamiento de un ordenador es más mecánico por el contrario de los humanos, por lo que su principal obstáculo radica en el gasto de recursos, que por el momento, se enfoca en la memoria y en el tiempo que se ha de tardar para que sea desarrollado el ordenamiento.

Dado lo anterior, se analizan cada algoritmo dado las características de su código, llegando a una función matemática que tiene como variable dependiente el tiempo de ejecución y com independiente la cantidad de archivos que ha de recibir.

En el manejo de Big Data es esencial contrar con algún orden para hallar elementos dentro los ficheros y es un previo al manejo de bases de datos como con SQL.

Muchos de los algoritmos de ordenamiento externos se basan en dividir el trabajo en varios sub procesos, por lo que el archivo original se ve modificado y se divide en partes ya sean iguales o diferentes dependiendo el algoritmo utilizado.

Los algoritmos de ordenamiento vistos en clase corresponden al ordenamineto polifásico, mezcla directa y una interpretación externa del conocido ordenamiento Radix. Los elementos a ordenar corresponden a listas o arreglos de números ya que involucrar demás caracteres supondría utilizar una jerarquía como el código ASCII utilizado en las terminales de los ordenadores MSDos para el llamado de caracteres específicos.

Por último se busca que los estudiantes aprendan acerca de la importancia de utilizar éstos métodos para empezar con el manejo de archivos en el lenguaje de programación JAVA y dándole un enfoque oriendtado a objetos con la interfaz gráfica que se ha implementado para poder realizar el ordenamiento de una mejor manera para la experiencia de usuario.

1. **Naturaleza del proyecto**
   1. **Antecedentes**

El ordenamiento de archivos es primordial para realizar muchos otros algoritmos como la búsqueda de manera eficiente. Existen muhcos tipos de ordenamiento y empezando por su principal separación: ordenamiento interno u ordenamineto externo. El ordenamiento interno se basa en lograr ordenar un elemento que se encuentre en la memoria principal, el ordenamiento interno por contraparte se refiere a ordenar elementos dentro de un fichero de la memoria secundaria mejor conocido como almacenamiento en disco del ordenador. El segundo es a lo que se enfoca ésta implementación con los algoritmos de tipo polifase, mezcla equilibrada y radix como anfitriones de la realización de todos los procesos correspondientes como la creación de archivos auxiliares para lograr simular el verdadero proceso del ordenamiento de un archivo de grandes dimensiones, haciendo particiones del total de elementos que hay y utilizando los archivos auxiliares para que así pueda ser procesador de la mejor manera

* 1. **Planteamiento del problema**

Se tiene un archivo de texto plano de formato .txt que se plantea como un prototipo de un enorme archivo el cual debe ser separado en secciones ya que por su tamaño es poco conveniente que la memoria se esfuerce leyendo esa cantidad de elementos uno por uno. Los métodos que se han de utilizar para desarrolar ésta tarea son diversos aunque entre más veloz sea el algoritmo consume una mayor cantidad de memoria. Esto tiene un alcance en el acomodo de grandes archivos donde los elementos de éste deban ordenarse para poder relizar operaciones sobre el archivo ordenado, logrando implementar una búsqueda más eficinete que una búsqueda lineal como la binaria o incluso implementarse una tabla hash.

* 1. **Objetivo del proyecto**
* Identificar los algoritmos de ordenamiento externos planteados para poder analizarlos y desarrollarlos.
  1. **Actividades**
* Elaborar un algoritmo que implemente el acomodo por el método POLIFASE y demostrar su funcionamiento
* Elaborar un algoritmo que implemente el acomodo por el método MEZCLA EQUILIBRADA y demostrar su funcionamiento
* Elaborar un algoritmo que implemente el acomodo por el método RADIX y demostrar su funcionamiento.
* Enfocar el trabajo a una programación orientada a objetos con uso de interfaz gráfica.

1. **Marco teórico**
   1. **Ordenamiento Polifásico.**

El ordenamiento polifásico se basa principalmente en el uso de un número asignado por el usuario para crear su elementos de ese tamaño e ir juntando 2 para ordenarlos de poco en poco.

* 1. **Ordenamiento por Mezcla Equilibrada**
  2. **Ordenamiento por Radix**

Es un algoritmo de ordenamiento estable que considera a las claves como secuencias de datos con diferentes valores posibles. Así, para ordenar las claves, el método las ordena por cada uno de sus datos, para números considera los dígitos y para las cadenas considera cada carácter. A cada ordenación de las claves por un dato se le llama iteración.

Hay dos tipos de clasificación de Radix:

MSD Radix Sort comienza a ordenar desde el digito más significativo (el digito que se encuentra más a la izquierda).

LSD Radix Sort comienza a ordenar desde el digito menos significativo (el digito que se encuentra más a la derecha)..

En este caso se analizará LSD Radix Sort externo con secuencias de dígitos.

Para realizar Radix Sort dada una lista de elementos enteros. Primero se crean los archivos auxiliares donde se almacenarán los elementos evaluados.

Posteriormente cada elemento es evaluado con respecto al digito menos significativo. En base a esté, el elemento es enviado al archivo correspondiente.

Una vez colocados todos los elementos dentro de los archivos, se vacía todo el contenido de cada archivo, comenzando con los que almacenan el digito de menor valor. Cada elemento sale en el orden en el que fue ingresado.

Este procedimiento se repite hasta que cada digito de los elementos sea evaluado.

1. **Desarrollo del proyecto**

**3.1 Ordenamiento por método Polifase**

**3.2 Ordenamiento por método de Mezcla Equilibrada**

**3.3 Ordenamiento por método Radix**

1. **Conclusiones**
2. **Anexos**
3. **Bibliografías**