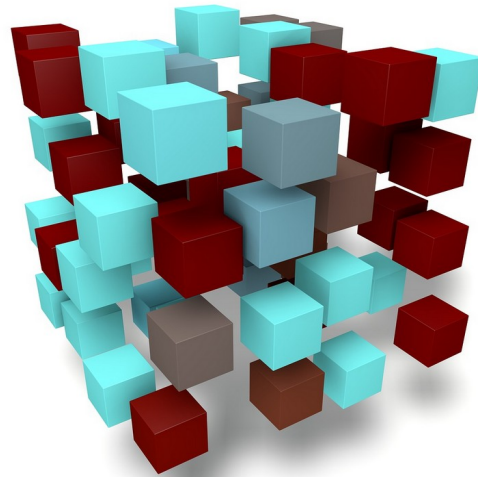


Estructura de Datos y Algoritmos II



Algoritmos de Ordenamiento.

Contenido.

- Objetivo.
- Repaso EDA I.
- Introducción.
- Ordenamiento.
- Ejemplo.
- Ejercicio.
- Fuentes.



Objetivo.

- General: El alumno será capaz de implementar lo aprendido el EDA II, y adquirirá conocimiento sobre temas diversos enfocados a las estructuras de datos.
- De la clase: El alumno comprenderá y aprenderá ciertos métodos de ordenamiento y será capaz de implementarlos.



Ejercicio de re paso.

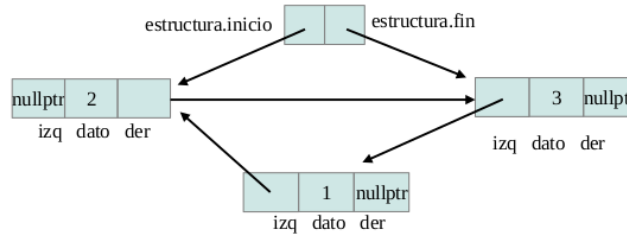
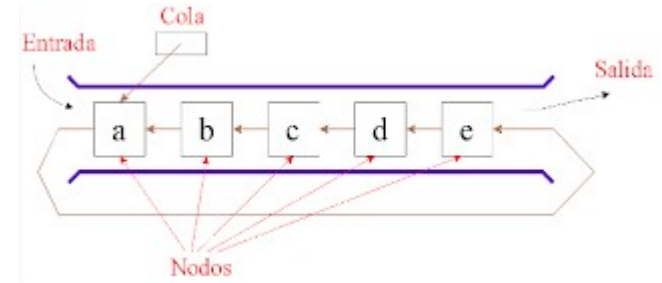
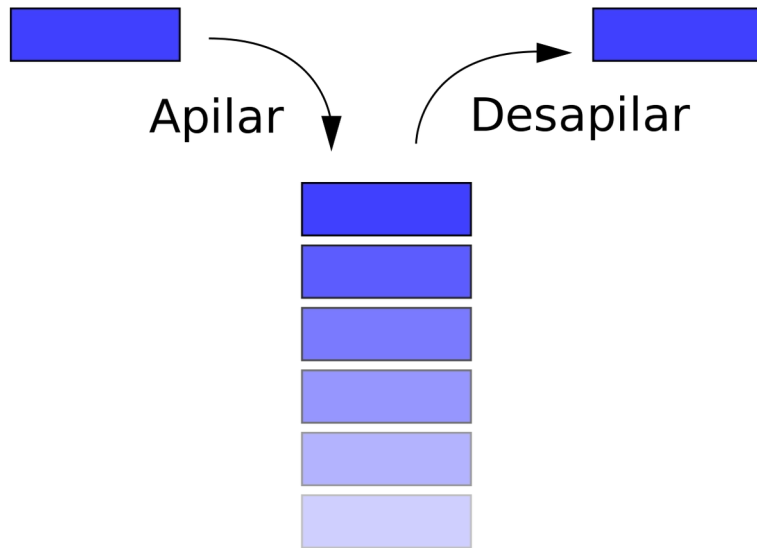
- Realiza el ejercicio que se ha publicado en Classroom.



Repaso.

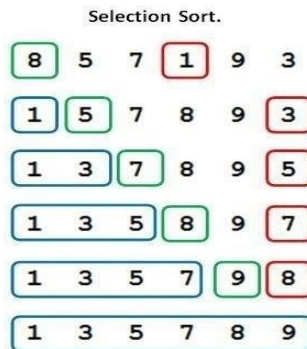
- Un *algoritmo* es un conjunto finito y ordenado de acciones con las que podemos resolver un problema.
- Un *problema* es una situación que se nos presenta y que, mediante la aplicación de un algoritmo tratamos de resolver.
- Para resolver los problemas, es necesario:
 - Analizarlo.
 - Diseñar el algoritmo.
 - Implementar dicho algoritmo en un lenguaje de programación.

Repaso.



Introducción.

- Ordenamiento o *Sorting*, se refiere a traer un conjunto de elementos a un lugar bien definido. Para ello, debemos especificar la noción de orden en los artículos a considerar.
- Por ejemplo, para los números podemos usar el orden numérico habitual (es decir, definido por la relación matemática "menor que" o "<") y para las cadenas la denominada lexicográfica u orden alfabético, que es el que usan los diccionarios y enciclopedias.
- Por lo general, lo que se entiende por ordenamiento u sorting es que una vez finalizado el proceso de clasificación, hay una forma sencilla de "visitar" todos los elementos en orden, por ejemplo, para imprimir el contenido de una base de datos.



Estrategias comunes de Ordenamiento.

- **Enumeration sorting:** Considere todos los elementos. Si sabemos que hay N ítems que son más pequeños que el que estamos considerando actualmente, entonces su posición final será en el número $N + 1$.
- **Exchange sorting:** Si se encuentran dos artículos fuera de servicio, cámbielos. Repita hasta que todos los elementos estén en orden.
- **Selection sorting:** Encuentre el elemento más pequeño, colóquelo en la primera posición, encuentre el elemento más pequeño de los elementos restantes, colóquelo en la segunda posición.
- **Insertion sorting:** Tome los elementos de uno en uno e insértelos en una estructura de datos inicialmente vacía de modo que la estructura de datos continúe ordenándose en cada etapa.
- **Divide and conquer:** Divida de forma recursiva el problema en subproblemas más pequeños hasta que tenga elementos individuales que sean triviales de ordenar. Luego, vuelva a juntar las "partes" clasificadas de manera que se conserve la clasificación.

¿Qué es ordenamiento?

- Es la operación de arreglar los registros de una tabla en algún orden secuencial de acuerdo a un criterio de ordenamiento.
- El ordenamiento se efectúa con base en el valor de algún campo en un registro. El propósito principal de un ordenamiento es el de facilitar las búsquedas de los miembros del conjunto ordenado.
- Ej. de ordenamientos:
 - Dir. telefónico, tablas de contenido, bibliotecas y diccionarios, etc.
- El ordenar un grupo de datos significa mover los datos o sus referencias para que queden en una secuencia tal que represente un orden, el cual puede ser numérico, alfabético o incluso alfanumérico, ascendente o descendente.
- ¿Cuándo conviene usar un método de ordenamiento?
 - Cuando se requiere hacer una cantidad considerable de búsquedas y es importante el factor tiempo.

Tipos de ordenamientos.

- **Internos:** Son aquellos en los que los valores a ordenar están en memoria principal, por lo que se asume que el tiempo que se requiere para acceder cualquier elemento sea el mismo ($a[1]$, $a[500]$, etc).
- **Externos:** Son aquellos en los que los valores a ordenar están en memoria secundaria (disco, cinta, cilindro magnético, etc), por lo que se asume que el tiempo que se requiere para acceder a cualquier elemento depende de la última posición de acceso (posición 1, posición 500, etc).



¿Qué algoritmos de ordenamiento son internos y cuáles externos?

- Internos:

- Inserción directa.
- Inserción binaria.
- Inserción directa.
- Selección directa.
- Selección directa.
- Burbuja.
- Shake.
- Intercambio directo.
- Shell.
- Inserción disminución incremental.
- Heap.
- Tournament.
- Ordenamiento de árbol.
- Quick sort.
- Sort particionado.
- Merge sort.
- Radix sort.
- Cálculo de dirección.

- Externos:

- Straight merging.
- Natural merging.
- Balanced multiway merging.
- Polyphase sort.
- Distribution of initial runs.

Ejemplificación de ordenamiento.

- Perlita tiene una colección impresionante de plumones y le gustaría ordenarlos para que pueda encontrarlos de forma más sencilla, pero también para que se vean bonitos. Perlita categoriza cada plumón con un color principal y un número que indica qué tan diluido está ese color, para hacer tonalidades más claras.
- Perlita quiere ordenar los plumones en el orden del arcoiris: Rojo, Anaranjado, Amarillo, Verde, Indigo, Violeta, Azul. Y dentro de cada color, usando el número del nivel de dilución, ordenarlos de menor a mayor.
- **Especificación de entrada**
 - La primera línea tiene un entero n , donde $1 < n < 500$ que representa el número de plumones que tiene Perlita. Después hay n líneas con un string representando el plumón s , este string tiene una de las palabras de los colores del arcoiris, listadas arriba, seguida de un - y un entero que puede ir desde el 0 hasta el 100. Ejemplos de plumones:
 - Anaranjado-43 Indigo-0 Rojo-100 Amarillo-43

Ejemplificación de ordenamiento.

- Especificaciones de salida
 - Imprime los plumones en el orden deseado, un plumón por línea.
- Ejemplo 1:
 - 5
 - Anaranjado-34
 - Indigo-3
 - Rojo-43
 - Indigo-0
 - Anaranjado-44
- Ejemplo 2:
 - Rojo-43
 - Anaranjado-34
 - Anaranjado-44
 - Indigo-0
 - Indigo-3

Ejercicio de ordenamiento.

- Resuelve el problema haciendo una implementación de quick sort o merge sort.
- El archivo ejecutable deberá ser adjuntado en el apartado de este ejercicio en Classroom, y deberá estar comentado el código.

NOTA: La solución de este ejercicio deberá ser realizada en el lenguaje de programación C o Python.

Fuente bibliográfica.

- Metodos de ordenamiento
- Data Structures and Algorithms



¡Gracias por su atención!

Antes que cualquier otra
cosa, la preparación es la
llave del éxito.

Alexander Graham Bell

