

**TRABAJO PRACTICO N°5**  
**Tema: Ordenación**

---

1. Dado un arreglo con los valores: (23, 15, 23, 5, 6, 34, 7, 26, 9, 22, 76, 43, 0, 33, 7, 16) ordenarlo “a mano” utilizando los siguientes métodos:
  - a) Inserción.
  - b) Selección.
  - c) Burbuja.
  - d) Burbuja Mejorado.
  - e) Mergesort.
  - f) Heapsort.
  - g) Quicksort.
  - h) Shellsort.
  - i) Shakersort.
  
2. Dado un arreglo con los valores: (23, 15, 2, 5, 6, 11, 19, 14, 20, 8, 17, 3, 22, 9, 12, 18, 1, 16, 4, 7, 10, 13, 21) ordenarlo “a mano” utilizando el método Binsort.
  
3. Hacer lo siguiente:
  - a) Dado un arreglo con los valores: (113, 41, 300, 985, 462, 249, 101, 914, 130, 747, 99, 159, 851, 238, 262, 746, 413, 14, 960, 313, 736, 165) ordenarlo “a mano” utilizando el método Radixsort.
  - b) Dada la lista de palabras en Inglés: (LOW, DOG, SEA, RUG, ROW, MOB, BOX, TAB, BAR, EAR, TAR, DIG, BIG, TEA, NOW, FOX) ordenarla “a mano” utilizando el método Radixsort. Luego, probarlo en la máquina.
  
4. Decir de qué manera los métodos de ordenación anteriores se ven afectados por el estado original de los datos:
  - a) Los datos están completamente desordenados.
  - b) Los datos están ordenados de menor a mayor.
  - c) Los datos están ordenados de mayor a menor.
  - d) Los datos están ordenados con un solo elemento fuera de lugar.
  
5. Implementar los siguientes métodos de ordenación y evaluar la cantidad de comparaciones y la cantidad de intercambios necesarios para ordenar arreglos aleatorios de distinto tamaño N (una tabla para comparaciones y otra para intercambios):

N	100	200	500	1000	2000
Mergesort					
Heapsort					
Quicksort					
Shellsort					
Shakersort					

6. En los algoritmos Burbuja o Selección, el elemento más pequeño de  $vec[i...n]$  es colocado en la posición  $i$  mediante intercambios, para valores sucesivos de  $i$ . Otra posibilidad es colocar el elemento máximo de  $vec[0...j]$  en la posición  $j$ , para valores de  $j$  entre  $n-1$  y  $0$ . A este algoritmo se le denomina ordenación por Ladrillos (Bricksort). Implementar dicho algoritmo y estudiar su complejidad.
  
7. Una variante curiosa de los algoritmos anteriores resulta al combinar los métodos de la Burbuja y de los Ladrillos. La idea es ir colocando alternativamente el mayor valor de  $a[0...j]$  en  $a[j]$ , y el menor valor de  $a[i...n-1]$  en  $a[i]$ . Implementar dicho algoritmo, conocido como Sacudidas (Shakersort).
  
8. Dado un vector cualquiera, que contiene objetos con distintos atributos, defina un algoritmo que implemente el método Mergesort sobre el vector, para cualquiera de los atributos del mismo. El método debe

## **TRABAJO PRACTICO N°5** **Tema: Ordenación**

---

aceptar como parámetros el vector, el número del atributo por el cual se quiere ordenar (clave) y el orden que se desea (ascendente o descendente).

**9.** Es posible utilizar la estructura de datos conocida como Heap (implementación usual para Colas de Prioridad) para ordenar un vector de  $n$  elementos.

- a) Implementar el método ordenación basado en esta estructura.
- b) Calcule el orden de ejecución para el peor caso del armado de un Heap a partir de un arreglo inicialmente desordenado.

**10.** Modifique el método Quicksort para que tome como pivote:

- a) El primer elemento.
- b) El último elemento.
- c) La mediana entre el primer, el del medio y el último elemento.
- d) Usando dos pivotes cualesquiera ( $\text{pivote1} < \text{pivote2}$ ).

**11.** Se tienen tres listas doblemente enlazadas de los estudiantes de las tres materias del área de Programación, las tres listas se encuentran ordenadas por DNI. Elabore un algoritmo en Java que permita, crear una nueva lista de todos los estudiantes del área, que se encuentre ordenada por DNI, deberá aplicar el método de Shellsort para crear la nueva lista ordenada.

**12.** Se dice que un método de ordenación es estable si al final del mismo, elementos de igual valor permanecen en el mismo orden en que se encontraban originalmente en el conjunto. Determinar cuáles de los todos los métodos vistos son estables. Explique cuando considere necesario para cual implementación es estable y para cual no.

**13.** Dada una secuencia  $a_1, a_2, \dots, a_n$  de enteros distintos, se desea determinar el  $k$ -ésimo elemento, esto es, aquel elemento  $a_i$  en la secuencia tal que existen exactamente  $k-1$  elementos menores que él en la secuencia. Una forma obvia de resolver el problema es ordenar la secuencia en forma ascendente, y tomar el elemento que queda en la posición  $k$ . Otra forma posible, es basándose en la idea de algoritmos de ordenación, definir un algoritmo que determine el  $k$ -ésimo elemento sin ordenar totalmente la secuencia.

**14.** En un archivo se ha guardado la lista de pasajeros de un vuelo con salida de Río Gallegos. Se sabe que el número de pasajeros no sobrepasa los 350 y que cada pasajero está identificado con un número de control en el rango de 1 a 999. Por un error informático hay números de control repetidos. Escribir un programa que realice las siguientes operaciones:

- a) Ordenar la lista de pasajeros por el número de control, usando Binsort.
- b) Los pasajeros que tienen un número de control repetido asignarles un número de control no existente.
- c) Pasar al archivo la lista de pasajeros en orden creciente con respecto a su número de control.

**15.** La clase Fecha (fecha de nacimiento de una persona) contiene los atributos día, mes y año. Escribir un programa que tenga como entrada el nombre y la fecha de nacimiento de los alumnos de un colegio. La salida ha de ser un listado en orden de nacimientos. Utilizar como método de ordenación Radixsort.