

TECNICATURA UNIVERSITARIA EN
PROGRAMACIÓN A DISTANCIA



PROGRAMACIÓN I

COMISIÓN 2
ESTUDIANTES

BÚSQUEDA Y ORDENAMIENTO

Chiaravalloti Agustín
Etchevest Jorgelina Carla

PROFESORA: JULIETA TRAPÉ
TUTOR: MIGUEL BARRERA OLTRA

Algoritmos de búsqueda y ordenamiento

- Clave para manejar grandes cantidades de datos de forma rápida y eficiente.
- Python para estudiar y comparar distintos métodos, analizando cuándo y cómo conviene aplicarlos.

Algoritmos de búsqueda Lineal

Recorre una lista elemento por elemento hasta encontrar el valor buscado. Es fácil de implementar y funciona en listas no ordenadas, pero resulta ineficiente en listas grandes ya que no aprovecha ningún tipo de orden.

Busqueda Binaria

eficiente en listas ordenadas, ya que divide el espacio de búsqueda a la mitad en cada paso. Es mucho más rápida que la búsqueda lineal, pero solo funciona si los datos están previamente ordenados y su implementación es un poco más compleja.

Ordenamiento

organizan datos según un criterio, como de menor a mayor. Métodos simples como burbuja son fáciles pero lentos, mientras que otros más avanzados como QuickSort o MergeSort son más eficientes y adecuados para grandes volúmenes de datos.



Mergesort

Divide la lista en partes pequeñas y las fusiona ordenadamente. Usa división y conquista para ordenar de forma eficiente.

Quicksort

Elige un pivote, divide la lista en menores y mayores, y repite el proceso recursivamente

RESULTADOS OBTENIDOS

- El rendimiento de QuickSort mejoró al usar la mediana de tres como pivote, evitando desbalances y recursividad excesiva.

CONCLUSIONES

- Elegir bien el algoritmo y su implementación según los datos es clave para lograr eficiencia y buen rendimiento.



**MUCHAS
GRACIAS!**

JUNIO 2025