Tecnicatura en Programación a Distancia - UTN

Trabajo Práctico Integrador

PROGRAMACIÓN I

Integrantes: Macarena Cantoni, Santiago Caiciia Massello

Profesor: Ariel Enferrel

Algoritmos de Búsqueda y Ordenamiento

Algoritmos de búsqueda

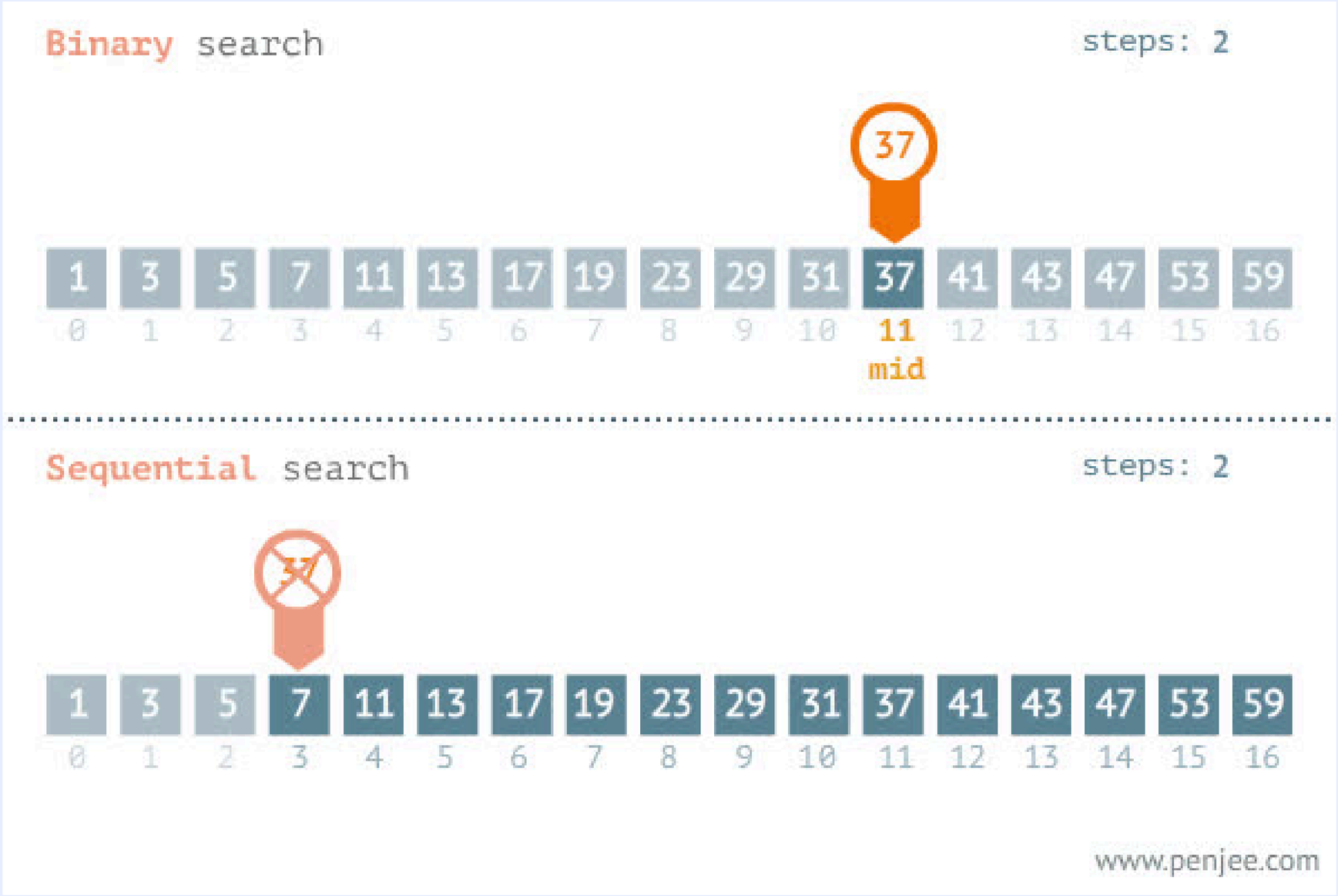
Lineal

- Itera a través de la lista de elementos uno por uno hasta encontrar el objetivo.
- Puede aplicarse a cualquier tipo de lista, ordenada o no.
- No es eficiente en listas grandes.

Binaria

- Divide la lista en dos partes y busca en la mitad correspondiente.
- Funciona en listas ordenadas.
- Realiza menos comparaciones.

Comparación



Medición

Notación O(n):

Algoritmos de ordenamiento

Bubble Sort

- Compara cada elemento de la lista con el siguiente y los intercambia si están en el orden incorrecto.

Selection Sort

- Encuentra el elemento más pequeño de la lista y luego lo intercambia con el primer elemento. Este proceso se repite hasta que todos los elementos de la lista estén ordenados.

Algoritmos de ordenamiento

Insertion Sort

- Inserta cada elemento de la lista en su posición correcta en la lista ordenada.

Quick Sort

- Divide la lista en dos partes y luego ordenando cada parte de forma recursiva.

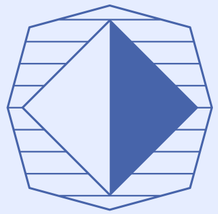
Merge Sort

- Divide la lista en dos partes, ordenando cada parte y luego fusionando las dos partes ordenadas.

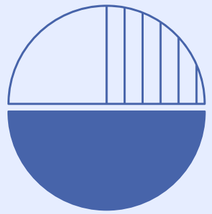
¿Por que es importante el uso?

- Eficiencia
- Organización
- Escalabilidad
- Precisión
- Versatilidad

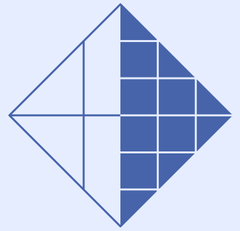
Metodología



Modularización



Uso de métodos de búsqueda y ordenamiento



Uso de librerías como random, os y time

Resultados

Quicksort

Demostró ser significativamente más rápido que el algoritmo de burbuja, especialmente en listas grandes.

Búsquedas

El algoritmo binario fue mucho más eficiente que la búsqueda secuencial, pero solo cuando la lista estaba previamente ordenada.

Búsqueda secuencial

Aunque más lenta, resultó útil cuando la lista estaba desordenada.

Medición

Se midieron tiempos de ejecución con la función time de Python, lo que permitió comparar empíricamente el rendimiento de cada algoritmo.

Conclusiones

- La elección del algoritmo de búsqueda u ordenamiento adecuado puede tener un impacto significativo en el rendimiento del programa.
- Consideraciones al momento de elegir el método indicado.
- Conceptos teóricos afianzados.