

ANÁLISIS Y SIMULACIÓN DEL ALGORITMO DE BÚSQUEDA BINARIA

DOSSON ARNOLDO LUQUES
NARVAEZ

Introducción

En el desarrollo de sistemas informáticos modernos que requieren manejo eficiente de grandes volúmenes de datos, como aplicaciones de inventarios empresariales, motores de búsqueda web, bases de datos relacionales y sistemas de gestión de información, surge constantemente la necesidad de localizar información específica de forma rápida y precisa. El uso de algoritmos inadecuados para esta tarea fundamental puede causar demoras significativas en la ejecución de aplicaciones, consumo excesivo de recursos computacionales, y en casos críticos, la degradación completa del rendimiento del sistema.



OBJETIVOS DEL PROYECTO

General

Evaluar el rendimiento teórico y empírico del algoritmo de búsqueda binaria mediante análisis comparativo.

Especificos

- Describir el funcionamiento interno del algoritmo
- Analizar su complejidad computacional (mejor/peor/promedio)

Especificos

- Medir desempeño en tiempo y memoria
- Comparar eficiencia vs búsqueda lineal
- Validar experimentalmente las fórmulas teóricas

MARCO CONCEPTUAL

BÚSQUEDA BINARIA:

- Algoritmo de "divide y vencerás"
- Reduce espacio de búsqueda a la mitad en cada iteración
- Requiere datos previamente ordenados

COMPLEJIDAD ALGORÍTMICA:

- Mejor caso: $O(1)$ – elemento en el centro
- Peor caso: $O(\log n)$ – múltiples divisiones
- Caso promedio: $O(\log n)$



ANÁLISIS TEÓRICO

EFICIENCIA ESPACIAL: $O(1)$

- Solo requiere variables de control
- Sin estructuras adicionales
- Memoria constante independiente del tamaño

EFICIENCIA TEMPORAL: $O(\log n)$

- Operaciones elementales por iteración: 8
- Fórmula: $T(n) = 6 + 8 \times \log_2(n)$
- Crecimiento logarítmico confirmado

Tamaño (n)	Tiempo (μ s)	$\log_2(n)$	Operaciones	Pasos
100	8.03	6.64	65.0	7.0
200	19.46	7.64	74.0	8.0
400	71.70	8.64	83.0	9.0
600	89.25	9.23	87.0	9.0
800	103.41	9.64	91.0	10.0
1000	121.68	9.97	93.0	10.0

RESULTADOS - GRAN ESCALA

Tamaño (n)	Tiempo (μs)	$\log_2(n)$	Pasos	Memoria (KB)
10,000	107.50	13.29	14	351.7
20,000	125.34	14.29	15	632.3
40,000	142.67	15.29	16	1264.6
60,000	156.89	15.87	16	1896.9
80,000	168.45	16.29	17	2529.2
100,000	179.23	16.61	17	3161.5

- Tiempo se mantiene bajo (<200 μs)
- Solo 3 pasos adicionales de 10K a 100K elementos
- Memoria crece linealmente con datos, no con algoritmo

COMPARACIÓN BINARIA VS LINEAL

Tamaño	Binaria (μs)	Lineal (μs)	Ventaja	Pasos Bin	Pasos Lin	Mejora
1,000	111.80	2,502.10	22.4x	10	1,000	100.0x
5,000	97.70	14,438.80	147.8x	13	5,000	384.6x
10,000	107.50	30,541.40	284.1x	14	10,000	714.3x
25,000	140.90	75,292.80	534.4x	15	25,000	1666.7x
50,000	178.10	153,799.00	863.6x	16	50,000	3125.0x
100,000	145.60	300,250.00	2062.2x	17	100,000	5882.4x

- En listas grandes: >2,000 veces más rápida
- Ventaja crece exponencialmente con el tamaño
- Diferencia crítica para aplicaciones reales

ANÁLISIS DE MEMORIA

Consumo de Memoria RAM

Tamaño (n)	Lista (KB)	Variables (KB)	Total (KB)
1,000	27.3	0.0	27.4
5,000	136.7	0.0	136.7
10,000	273.4	0.0	273.5
20,000	546.9	0.0	546.9
50,000	1367.2	0.0	1367.2
100,000	2734.4	0.0	2734.4

Memoria (bytes)

- Variables del algoritmo permanecen constantes
- Porcentaje tiende a cero conforme crecen los datos
- Extremadamente eficiente en uso de memoria

APLICACIONES PRÁCTICAS

DONDE SE USA BÚSQUEDA BINARIA:

- Bases de datos (índices B-tree)
- Motores de búsqueda (algoritmos de ranking)
- Sistemas operativos (gestión de memoria)
- Videojuegos (búsqueda de elementos)
- Aplicaciones financieras (búsqueda de transacciones)
- Sistemas de inventario
- Bibliotecas digitales



CONCLUSIONES PRINCIPALES

1. SUPERIORIDAD CONFIRMADA:

- 2,000+ veces más rápida en listas grandes
- Diferencia crece exponencialmente

2. EFICIENCIA ESPACIAL ÓPTIMA:

- $O(1)$ - memoria constante confirmada
- <0.01% overhead en listas grandes

3. VALIDACIÓN TEÓRICA:

- Fórmulas matemáticas precisas
- Diferencia real-teórico < 1 paso

4. APLICABILIDAD PRÁCTICA:

- Ideal para datos ordenados estáticos
- ROI positivo con >20-31 búsquedas

MUCHAS GRACIAS

