

# Algoritmo Búsqueda Galopante o Búsqueda Exponencial

Benjamin Alejandro Mosso Miller  
Diego Alexis Salazar Jara

29 de mayo de 2025

## 1. Problema 1

Sea  $A$  un arreglo de enteros ordenados de menor a mayor. El algoritmo *BusquedaBinaria*( $A, li, ls, x$ ) permite verificar si  $x \in A$  en tiempo  $O(\log_2 n)$ . La Búsqueda Galopante (BG) consiste de dos etapas. En la primera se determina un rango donde la clave  $x$  debería encontrarse (si es que  $x \in A$ ); y en la segunda se realiza una búsqueda binaria en el rango determinado en la primera etapa (Ver Algorithm 1). El límite inferior ( $lin\ f$ ) del rango corresponde a la mayor posición (o mayor valor del índice de  $A$ ) tal que  $A[lin\ f] < x$  y que es potencia de 2 (ver líneas 6-8 de Algorithm 1).

---

```

1  BusquedaGalopante(A, size, x)
2  if (size = 0) then
3      return -1
4  else
5      lin f = 1
6      while (lin f * 2 < size and A[2 * lin f] < x) do
7          lin f = lin f * 2
8      end while
9      return BusquedaBinaria(A, lin f, size, x)
10 end if

```

---

Algoritmo 1: Búsqueda Galopante o Exponencial

1. Implemente en su lenguaje favorito la Búsqueda Binaria y la Búsqueda Galopante.
2. Genere al azar un arreglo  $A$  de tamaño 5.000.000.000 de enteros. El rango de los elementos de  $A$  es  $[1, 10.000.000]$ . Ordene el arreglo  $A$  (puede usar Merge Sort u otro). Luego genere al azar las cantidades de claves: 1.000, 5.000, 10.000, 50.000 y 100.000 y complete la tabla siguiente:

Número de Claves	Búsqueda Binaria Tiempo (ms)	Búsqueda Galopante Tiempo (ms)
1.000		
5.000		
10.000		
50.000		
100.000		

Tabla 1: Comparación Algoritmos de Búsqueda Binaria y Galopante.



3. De acuerdo a los datos de la tabla anterior, ¿Cuál algoritmo es mejor?.
4. Justifique haciendo un estudio teórico de ambos algoritmos.