

ALGORITMOS DE BÚSQUEDA

1. Secuencial o Lineal
2. Secuencial o Lineal con Centinela
3. Binaria o dicotómica

[Ejercicios](#)

Definición

- Un algoritmo de búsqueda es aquel que está diseñado para localizar un elemento con ciertas propiedades dentro de una estructura de datos; por ejemplo, en vectores o matrices.
- Consiste en recorrer la estructura de datos utilizando diferentes técnicas o recorridos a fin de datos con el valor buscado.
- Los procesos de recorrido varían, desde simples recorridos lineales desde el inicio del algoritmo, hasta la utilización de técnicas más eficientes, como la búsqueda binaria.
- Algunos algoritmos de búsqueda necesitan aplicar Algoritmos de Ordenamiento primero, a fin de producir resultados correctos.

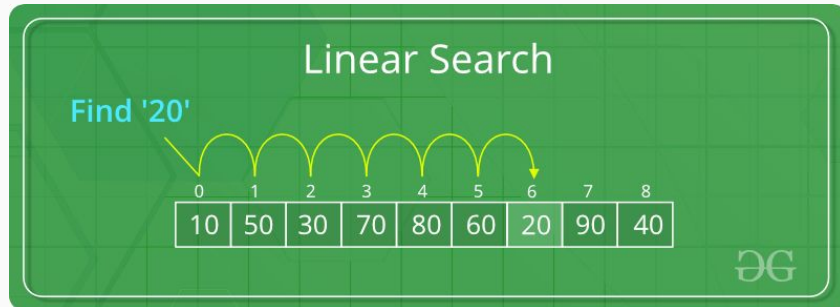
Definición

- Se utiliza cuando el vector no está ordenado o no puede ser ordenado previamente.
- Consiste en buscar el elemento comparándolo secuencialmente (de ahí su nombre) con cada elemento del vector hasta encontrarlo, o hasta que se llegue al final.
- La existencia se puede asegurar cuando el elemento es localizado, pero no podemos asegurar la no existencia hasta no haber analizado todos los elementos del vector.
- Permite recuperar las posiciones de varias ocurrencias del valor buscado.

ALGORITMOS DE BÚSQUEDA - Secuencial o Lineal

Proceso de la Búsqueda Secuencial

- Se utiliza un ciclo iterativo y se comienza a recorrer el arreglo hasta encontrar el valor buscado.
- Retorna el índice del arreglo en donde se encuentra el valor buscado.
- Se sigue recorriendo hasta el final, si se encuentra otro valor igual al buscado, se actualiza el resultado de la búsqueda.



Implementación de la Búsqueda Secuencial o Lineal

```
public static int busquedaLineal(int[] arreglo, int valor) {  
  
    int i = 0;  
    int pos = -1;  
  
    while(i < arreglo.length) {  
        if (arreglo[i] == valor) {  
            pos = i;  
        }  
        i++;  
    }  
  
    return pos;  
}
```

Definición

- Se utiliza cuando el vector no está ordenado o no puede ser ordenado previamente.
- Se agrega un centinela que permite conocer el estado del proceso y permite finalizar el recorrido en caso de encontrar el elemento.
- Consiste en buscar el elemento comparándolo secuencialmente (de ahí su nombre) con cada elemento del vector hasta encontrarlo, o hasta que se llegue al final.
- Se puede asegurar la no existencia de un elemento, debido a que de existir el valor, el centinela finaliza el proceso.
- Se optimiza cuando el vector está ordenado.
- Retorna la primera ocurrencia de un valor.

Implementación de la Búsqueda Lineal con Centinela

```
private static int busquedaLinealCentinela(int[] arreglo, int valor) {  
  
    int i = 0;  
  
    while(i < arreglo.length && arreglo[i] != valor) {  
        i++;  
    }  
  
    if (i < arreglo.length)  
        return i;  
  
    return -1;  
}
```

Definición

- Se utiliza cuando el vector en el que queremos determinar la existencia de un elemento está previamente ordenado.
- Este algoritmo reduce el tiempo de búsqueda considerablemente, ya que disminuye exponencialmente el número de iteraciones necesarias.
- Para implementar este algoritmo se compara el elemento a buscar con un elemento cualquiera del vector (normalmente el elemento central): si el valor de éste es mayor que el del elemento buscado se repite el procedimiento en la parte del vector que va desde el inicio de éste hasta el elemento tomado, en caso contrario se toma la parte del vector que va desde el elemento tomado hasta el final.
- De esta manera obtenemos intervalos cada vez más pequeños, hasta que se obtenga un intervalo indivisible. Si el elemento no se encuentra dentro de este último entonces se deduce que el elemento buscado no se encuentra en todo el vector.

Proceso de la Búsqueda Binaria

1. Dado un arreglo ordenado.
2. Se realiza el proceso iterativo de encontrar el medio del arreglo.
3. Validar si el valor buscado se encuentra en el medio.
4. Si no se encuentra, se procede a decidir qué lado del arreglo deberá ser descartado.
5. Se regresa al punto 2 y continúa hasta dar con el valor buscado.



1	2	3	9	11	13	17	25	57	90
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]

Implementación de la Búsqueda Binaria

```
private static int busquedaBinaria(int[] arreglo, int valor) {  
    int min = 0;  
    int max = arreglo.length - 1;  
    int medio = 0;  
  
    boolean encontrado = false;  
  
    while (min <= max && !encontrado)  
    {  
        medio = (min + max) / 2;  
        if (arreglo[medio] == valor)  
            encontrado = true;  
        else  
        {  
            if (arreglo[medio] > valor)  
                max = medio - 1;  
            else  
                min = medio + 1;  
        }  
    }  
  
    if (!encontrado)  
        return -1;  
  
    return medio;  
}
```

Comparaciones entre Búsquedas Lineales y Binarias

- En el ejemplo se busca el valor **37**
- Se cuenta la cantidad de iteraciones que se requieren para encontrar el valor en cada caso.

