



Algoritmos de Búsqueda

Julio, 2013
Guadalajara, Jalisco. México



Autor
Prof. Sabrina Lizbeth Vega Maldonado

Algoritmos de Búsqueda

Un algoritmo de búsqueda es un método cuya función es localizar un dato.

Dado un conjunto de datos almacenados en un arreglo, determinar cuál es la posición en la que se encuentra el elemento buscado o en su defecto, determinar que no se encontró en ese conjunto.

Algoritmos de Búsqueda

A continuación se presentan 2 algoritmos para realizar búsquedas:

- ◆ Búsqueda Secuencial

- ◆ Búsqueda Binaria

Secuencial



90	30	50	42	67	23
----	----	----	----	----	----

Cuando se realiza un búsqueda secuencial sobre un arreglo se comparan uno a uno todos los elementos del arreglo hasta que encuentra el que esta buscando.

Si suponemos que en el arreglo **L** se busca el 67, la búsqueda secuencial va a comparar con todos los elementos hasta encontrar al 67

Secuencial

90

30

50

42

67

23

Secuencial (67, L) : Realiza 5 comparaciones para lograr encontrar al 67

Secuencial (77, L) : Realiza 6 comparaciones para darse cuenta que el 77 no esta

Secuencial (90, L) : Realiza 1 comparación para lograr encontrar el 90

Secuencial (23, L) : Realiza 6 comparaciones para encontrar el 23

Secuencial

90

30

50

42

67

23

Secuencial (90, L) : Realiza 1 comparación para lograr encontrar el 90

A la búsqueda del 90 es a lo que llamamos el **mejor de los casos** porque al primer intento lo encuentro

Secuencial (23, L) : Realiza 6 comparaciones para encontrar el 23

A la búsqueda del 23 es a lo que llamamos el **peor de los casos** porque hasta el último intento lo encuentro

Busqueda Secuencial

```
int secuencial (int numeros[], int tam, int x){  
    for (i=0; i<tam; i++) {  
        if ( x == numeros [ i ] ) {  
            return i ;  
        }  
    }  
    return -1;  
}
```

Binaria

90	30	50	42	67	23
23	30	42	50	67	90
0	1	2	3	4	5

Antes de realizar una búsqueda binaria sobre un arreglo, se necesita que el arreglo esté ordenado. Entonces va dividiendo el arreglo por la mitad hasta que encuentra el que está buscando.

Binaria

23	30	42	50	67	90
0	1	2	3	4	5

Si suponemos que busca el 50 va a dividir el arreglo por la mitad

23	30	42	50	67	90
-----------	-----------	-----------	-----------	-----------	-----------

Descarta la primer mitad y vuelve a dividir el arreglo por la mitad

50	67	90
-----------	-----------	-----------

Descarta la segunda mitad y vuelve a dividir el arreglo por la mitad

50

Binaria

23

30

42

50

67

90

Binaria (50, L) : Realiza 3 comparaciones para lograr encontrar al 50

Binaria (42, L) : Realiza 1 comparación para lograr encontrar el 42

Binaria (30, L) : Realiza 3 comparaciones para encontrar el 30

Busqueda Binaria

```
int binaria (int numeros[], int tam, int x){
    int inicio = 0;
    int fin = tam-1;
    int medio;
    do {
        medio = ( inicio + fin ) / 2 ;
        if ( x == numeros [ medio ] ) {
            return medio;
        }
        else if (x > numeros[medio]) {
            inicio = medio + 1;
        }
        else if (x < numeros[medio]) {
            fin = medio - 1;
        }
    } while ( inicio <= fin );
    return -1;
}
```

Secuencial & Binaria

23

30

42

50

67

90

Binaria (42, L) : Realiza 1 comparación para lograr encontrar el 42

A la búsqueda del 42 es a lo que llamamos el **mejor de los casos** porque al primer intento lo encuentro

Binaria (30, L) : Realiza 3 comparaciones para encontrar el 30

A la búsqueda del 30, 50 o el 90 es a lo que llamamos el **peor de los casos** porque requiere hacer el mayor número de comparaciones para encontrarlos

Ejercicio

10	17	6	33	5	21	42	8	3	22	15	13	1	55	50	9	18	7	25	2
----	----	---	----	---	----	----	---	---	----	----	----	---	----	----	---	----	---	----	---

Hacer la búsqueda secuencial y binaria del :

- 15
- 20
- 10

Escribir: número de comparaciones, resultado (posición), valor de inicio y fin para cada arreglo resultante