



11074 – Programación I

División Computación – Departamento de Ciencias Básicas



VECTORES: ALGORITMOS DE BÚSQUEDA

Métodos de Búsqueda y Ordenamiento

Búsqueda Secuencial

Una de las cosas que siempre necesitamos hacer sobre un arreglo es poder localizar un dato o saber si existe para poder realizar la tarea deseada.

Pues bien, pensemos que tenemos un arreglo de n elementos de tipo entero y que queremos saber cuál es la ubicación que ocupa un valor en particular dentro del arreglo.

¿Qué podría ocurrir al realizar esta búsqueda? Podemos pensar rápidamente dos posibilidades:

- 1 - Que el elemento a buscar esté en la lista
- 2 - Que el elemento a buscar, no esté en la lista.

Pero hay otras cosas que pueden pasar.

Por ejemplo, ¿qué pasa si el elemento a buscar está más de una vez? ¿Qué debería hacer el algoritmo en ese caso? Por otro lado, ¿qué pasa si el arreglo está vacío? ¿Es necesario buscar entonces? Si no lo reviso, ¿Daría error mi algoritmo?

Estas son algunas de las consideraciones que vamos a tener que plantearnos a la hora de resolver el problema.

Una vez que analizamos los posibles problemas que podemos tener, debemos analizar qué datos necesitamos para poder resolver el problema.

Uno de los datos que es seguro debemos tener es el **arreglo de n elementos**. Otro es el **número a buscar**. Otro dato que necesitamos es la **cantidad de elementos(n)** que posee el arreglo. Ahora vamos a tener que recorrer el arreglo y por tanto, vamos a tener que usar una **estructura iterativa**. Si sabemos con antelación cuántos elementos tiene el arreglo se usará **for**. Si no lo sabemos se usará **while** hasta que se cumplan las condiciones que establecimos para finalizar la búsqueda. También tenemos **do while** que es similar al while pero se ejecuta al menos una vez.

Cuál sería el algoritmo a confeccionar que nos permita realizar la tarea previamente planteada?

Tendríamos que recorrer el arreglo desde el principio, elemento por elemento y preguntar si es el elemento que estamos buscando. Si es así finaliza la búsqueda. Sino, hay que seguir buscando hasta encontrarlo o hasta que no haya más elementos.

Este algoritmo describe lo que conocemos como **Búsqueda secuencial**.

Este algoritmo describe lo que conocemos como **Búsqueda secuencial**.

Vamos a ver cómo lo desarrollamos en C (aunque se puede codificar en cualquier otro lenguaje de programación).

Definimos la función **búsqueda**:

```
int buscar(int clave, int datos[], int n){  
    int i;  
    i = 0;  
    while ( (i < n) && (datos[i] != clave) )  
        i = i + 1;  
    return i;  
}
```

En este ejemplo se muestra la implementación (en lenguaje C) del algoritmo de búsqueda secuencial en un arreglo de enteros (nótese que si no se encuentra el elemento buscado, la función buscar() retorna 'n', el tamaño del arreglo)

Búsqueda Binaria

El problema anterior se podría haber resuelto en forma más rápida si el arreglo hubiera estado ordenado en forma creciente o decreciente porque ya sabríamos dónde buscar.

Se basa en el método **DIVIDE y VENCERÁS**. El arreglo debe estar **ordenado**. Consiste en dividirlo en dos y comparar el elemento central con el valor a buscar, si son iguales termina la búsqueda, si no, se determina en qué mitad podría estar y se realiza el mismo procedimiento; así sucesivamente hasta encontrar el buscado o hasta no poder dividir más la lista.

```
int buscar(int clave, int datos[], int n) {
    int inferior, superior, centro, resu;
    inferior = 0;
    superior = n-1;
    resu= n;
    int encontrado = 0;
    while ( (inferior <= superior) && encontrado==0) {
        centro = (inferior + superior) / 2; /* Trunca hacia cero */
        if (clave == datos[centro]){
            resu = centro; /* Encontrado */
            encontrado = 1;}
        else {
            if (clave < datos[centro])
                superior = centro - 1; /* Buscar en la primera mitad */
            else
                inferior = centro + 1; /* Buscar en la segunda mitad */
        }
        /* Si no lo encuentra, retorna 'n' */
    }
    return resu;
}
```

// Enseñar no es transferir conocimientos, sino crear las posibilidades de su construcción; quien enseña aprende al enseñar y quien aprende enseña al aprender”



Paulo Freire