
uao

06

ESTRUCTURA DE DATOS Y ALGORITMOS 1

Algoritmos de Búsqueda

Profesor
Orlando Arboleda Molina

uao

Algoritmos de Búsqueda

- La **búsqueda** (**search**) consiste en **encontrar un elemento** (el mayor, menor o dato particular) en un arreglo, colección (u otra estructura de datos).

- Existen algoritmos de búsqueda, cada uno con su correspondiente complejidad, entre ellos:
  - **Linear Search** (búsqueda secuencial)
  - **Binary Search** (búsqueda binaria)

- La **complejidad** de la búsqueda **depende** de que los **elementos estén o no ordenados** y de la **estructura** en la cual han sido **almacenados**.

Ejemplo: si los datos están almacenados en un arreglo y este ya está ordenado, encontrar el mayor o menor es una operación eficiente, porque los datos se encuentran en la posición inicial o final. Siendo T(n) = O(1).

Linear Search

- Busca **recorriendo** cada uno de los **elementos posición a posición**, hasta que **se encuentra** el elemento buscado o se haya **recorrido toda los elementos**.

- Complejidad de la búsqueda secuencial
  - **Mejor caso –** que el elemento buscado se encuentre en la posición inicial, en cuyo caso:

$$T(n) = O(1)$$

  - **Peor caso** - que el elemento buscado se encuentre en la ultima posición recorrida o no se encuentra, en cuyo caso:

$$T(n) = O(n)$$

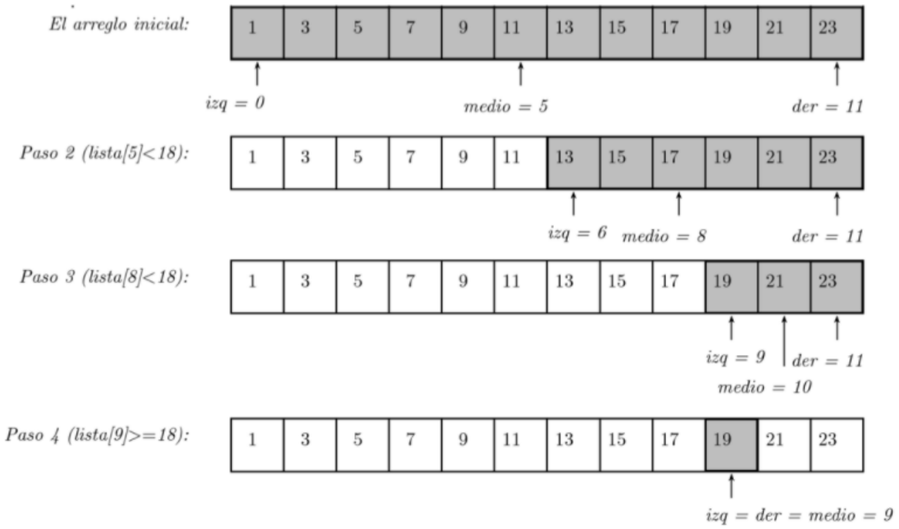
Reto: Conoce otro tipo de búsqueda que permita mejorar el tiempo y complejidad para encontrar un elemento, aunque en el peor caso?

Binary Search

- Requiere que el arreglo este ordenado.

- Busca el **elemento** en la **mitad** de la **secuencia**. **Si no** es el elemento buscado, solo **busca** en los **elementos s**upe**riores** (lado derecho) o en los **elementos inferiores** (lado izquierdo), según corresponda.

Ejemplo: comportamiento de la búsqueda binaria para encontrar al 18.



Binary Search

- **Análisis del peor caso:** cuando se realiza la mayor cantidad de comparaciones con el elemento de la mitad. (lo encontró en la ultima comparación posible o no se encuentra)

Versión iterativa

```
BINARY_SEARCH(A,p,q,r)
   m = (p+q)/2
   Mientras p<=q y A[m] diferente r
      Si r < A[m]
         q = m-1
      Sino
         p = m+1
      m = (p+q)/2
   FinMientras
   posicion = -1
   Si r igual a A[m]
      posicion = m
   Devolver posicion
```

Versión recursiva

```
BINARY_SEARCH(A,p,q,r)
   posicion = -1
   Si p <= q
      m = (p+q)/2
      Si r es igual A[m]
         posicion = m
      Sino
         Si r es menor a A[m]
            posicion = BINARY_SEARCH(A,p,m-1,r)
         Sino
            posicion = BINARY_SEARCH(A,m+1,q,r)
   Devolver posicion
```

$$T(n) = \begin{cases} O(1), & si \;\; n = 0 \; (p > q) \\ T(n/2) + O(1), & si \; n > 0 \; (p \leq q) \end{cases}$$

T(n) = T(n/2) + $O(1)$
T(n) = (T(n/4)+$O(1)$) + $O(1)$ = T(n/4) + 2*$O(1)$
…….
T(n) = $T(n/2^k)$+ k*$O(1)$

Se computara el valor de k (cuando se llega a l caso base)
Si $n/2^k = 1$ entonces $2^k = n$ por tanto k=log n

$$T(n) = T(n/2^k) + kO(1) \leq T(1) + (log\ n)\ O(1)$$
$$T(n) \leq O(1) + (log\ n)O(1) = O(log\ n)$$

Ejercicio de Implementación

- Realizar la **practica** correspondiente a **Búsqueda**. Para esta actividad, se debe disponer de:
  - El entorno del desarrollo **Visual Studio Code** y la extensión **Live Server**
  - Un **navegador moderno** y actualizado (ej. Chrome, Firefox, etc).

- Se solicita:
  - Incluir la implementación del método ***busquedaBinaria_por_placa*** basada en alguno de los pseudocódigos mostrados en clase (iterativo y recursivo), para que busque objetos de la clase Vehículos, con base en su placa.
  - Implementar el método ***buscarVehiculo*** para invoque la búsqueda realizada en el ítem previo y dependiendo, de si existe o no, despliegue el mensaje con el formato indicado en la práctica.
  - Implementar la lógica necesaria para que cuando se pulse el botón Buscar se despliegue el mensaje retornado por el método del ítem previo.

Reto: realizar la implementación del otro pseudocódigo binary search binaria presentado en clase y modificar el aplicativo para su ejecución.

06

BUEN VIENTO Y BUENA MAR !!!


uao