Semana 03

ESTRUCTURAS DISCRETAS: Algoritmos de Búsqueda

Mg. Flor Elizabeth Cerdán Léon

Mg. Flor Elizabeth Cerdán León

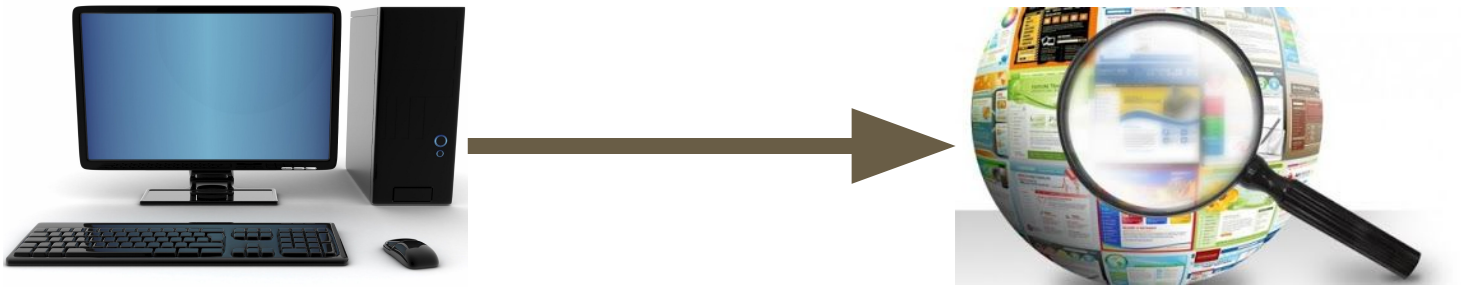
Introducción

Mg. Flor Elizabeth Cerdán León

...Introducción

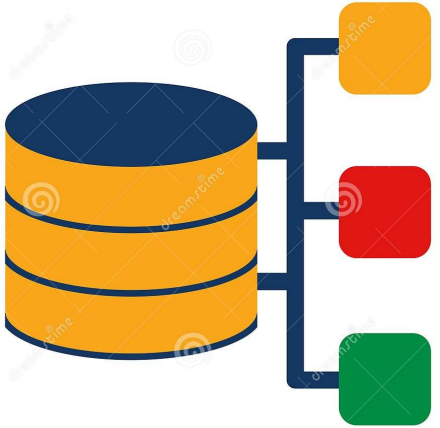
Mg. Flor Elizabeth Cerdán León

...Introducción



Mg. Flor Elizabeth Cerdán León

...Introducción

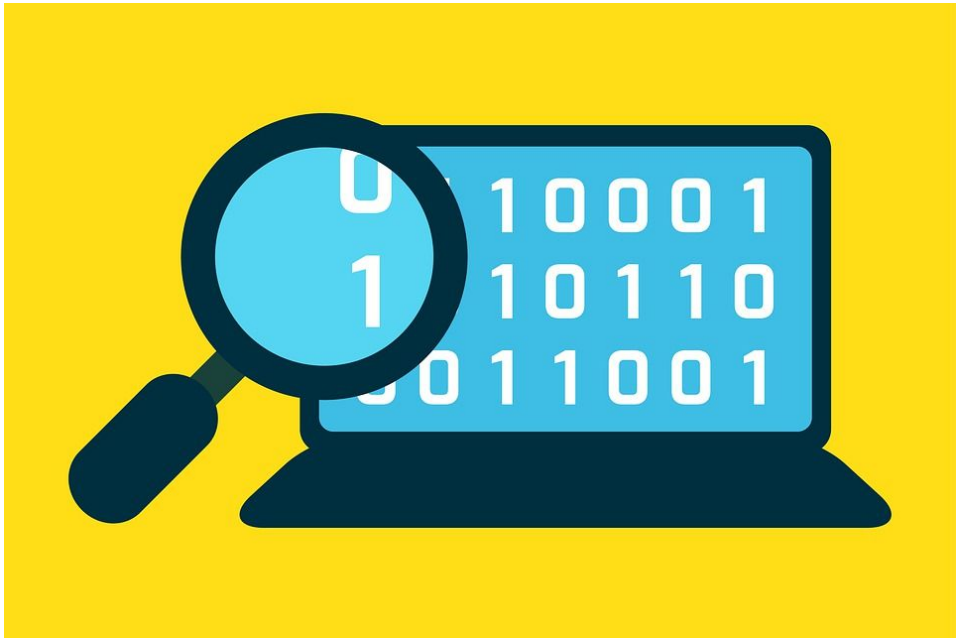


Algoritmo

- “Un conjunto de acciones para hallar la solución de un problema”.
- Los algoritmos se concibieron para resolver muchos problemas

Algoritmo de Búsqueda secuencial

- Es un método que consiste en ubicar un elemento de forma lineal dentro de una lista (No necesariamente ordenada).



...Búsqueda secuencial

Características

- Busca un elemento en una lista o arreglo no ordenado
- Los elementos se exploran secuencialmente (De forma lineal)
- Este algoritmo compara cada elemento del arreglo con el elemento buscado
- Es probable que el elemento a buscar sea el primero, el último o cualquier otro.

Ejemplo:

- Dado un arreglo llamado **lista**, busque el número 12.

Posición	0	1	2	3	4	5	6
Lista	12	18	-20	0	15	12	9

- Número a buscar: **12**
- Posición del número buscado = **0**

Algoritmo

```java
package busqueda;
// autor: Flor Elizabeth Cerdán León
import java.util.Scanner;
public class Secuencial {
    public static void main(String[] args) {
        Scanner teclado=new Scanner(System.in);
        int lista[]={12,18,-20,0,15,12,9,6};
        int num;
        System.out.print("ingresar valor a buscar:");
        num=teclado.nextInt();
        boolean encontrado=false;
        for (int i = 0; i < lista.length && encontrado==false; i++) {
            if(num==lista[i]) {
                encontrado=true;
                System.out.println("Posición del número buscado: " + i);
            }
        }
    }
}
```

Algoritmo de búsqueda binaria

Características

Es conocida como algoritmo de **búsqueda de intervalo medio** o **búsqueda logarítmica**.

Se aplica a una lista previamente **ordenada**

- **Primero**: Se analiza el punto medio del arreglo (el valor central), si es el valor buscado, se devuelve el índice del punto medio.
- **Segundo**: Si el valor buscado es mayor al valor del centro, se descarta el lado izquierdo de la lista.
- **Tercero**: Si el valor buscado es menor al valor del centro, se descarta el lado derecho de la

Ejemplo

Dado un arreglo llamado **Lista**, busque el número **29**.

Posición	0	1	2	3	4	5	6	7	8	9
Lista	2	4	5	9	12	15	16	28	29	51

...Ejemplo

Número a buscar: **29**

Inicio → 0

Último = Lista.length -1 → 9

1ra iteración:

Centro = (Inicio + Último) / 2 = (0+9)/2 → 4

ValorCentro = Lista[Centro] → 12

Primero: 29 no es igual a 12

Segundo: Como 29 es mayor a 12, se descarta el lado izquierdo

Entonces Inicio = Centro + 1 → 5

...Ejemplo

2da iteración:

Centro = (Inicio + Último) / 2 = (5+9)/2 → 7

ValorCentro = Lista[Centro] → 28

Primero: 29 no es igual a 28

Segundo: Como 29 es mayor a 28, se descarta el lado izquierdo

Entonces Inicio = Centro + 1 → 8

...Ejemplo

3ra iteración:

Centro = (Inicio + Último) / 2 = (8+9)/2 → 8

ValorCentro = Lista[Centro] → 29

Primero: 29 es igual a 29

Respuesta: Entonces la posición es 8

```java
package busqueda;
import java.util.Scanner;
public class Binario {
    public static void main(String[] args) {
        Scanner teclado=new Scanner(System.in);
        int lista[]={2,4,5,9,12,15,16,28,29,51};
        int num, centro, primero=0, ultimo, valorcentro;
        boolean encontrado = false;
        ultimo = lista.length-1;
        System.out.print("ingresar valor a buscar:");
        num=teclado.nextInt();
        while(primero <= ultimo && encontrado == false){
            centro = (primero + ultimo)/2;
            valorcentro = lista[centro];
            System.out.println("Comparando a " + num + " con " + valorcentro);
            if(num == valorcentro){
                encontrado = true;
                System.out.println(centro);
            }else if(num<valorcentro){
                ultimo = centro-1;
            }else{
                primero=centro+1;
            }
        }
        if(encontrado == false)
            System.out.println("Valor no encontrado");
    }
}
```