UNIVERSIDAD DE ALMERÍA

Grado en Ingeniería Informática
Metodología de la Programación

UAL. Área de Lenguajes y Sistemas Informáticos. Mercedes Martínez Durbán

Tema 0. Búsqueda y ordenación

Búsqueda

- Lineal o secuencial
- Binaria o dicotómica

Tema 0. Búsqueda y ordenación

- Búsqueda
- Ordenación

Tema 0. Búsqueda y ordenación

Búsqueda

❑ Lineal o secuencial

La búsqueda es un proceso que permite determinar si un valor específico está en un array. En el caso de una búsqueda lineal, compararemos el valor que buscamos y que llamaremos **clave,** secuencialmente con cada elemento o componente del array. El proceso continuará hasta que encontremos el valor buscado o hayamos visitado todos las componentes del array sin éxito. La búsqueda lineal devuelve la posición del elemento que coincide con la clave o -1 si no lo hemos encontrado.

Array a	4	6	8	-5	1
	[0]	[1]	.....		[4]

El valor **clave** se compara con a[i] utilizando un for i = 0,1,….

Ejemplos: clave = 4 ⇒ devuelve posición que ocupa en el array, 0.

clave = 50 ⇒ no está en el array, devuelve -1.

Tema 0. Búsqueda y ordenación

Búsqueda

- **Lineal o secuencial**

Ejemplo 1. Método de clase para la búsqueda secuencial.

```
/** El método busca el valor clave en el array a */
public static int busquedaLineal(int[] a, int clave) {
        for (int i = 0; i < a.length; i++) {
                if (clave == a[i])
                        return i;
        }
        return -1;
}
```

```java
package org.mp.tema00;

import java.util.Arrays;
import java.util.Scanner;

public class Busqueda1 {
    private static Scanner entrada;

    /** El método busca el valor clave en el array a */
    public static int busquedaLineal(int[] a, int clave) {
        for (int i = 0; i < a.length; i++) {
            if (clave == a[i])
                return i;
        }
        return -1;
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int[] a = { 4, 6, 8, -5, 1 };

        entrada = new Scanner(System.in);
        System.out.println("Introduce el valor que buscas en el array ");
        int clave = entrada.nextInt();
        int posicion = busquedaLineal(a, clave);
        if (posicion == -1) {
            System.out.println("El valor " + clave + " no está en el array");
            System.out.println(Arrays.toString(a));
        } else {
            System.out.println("El valor " + clave + " está en el array en la posición " + (posicion + 1));
            System.out.println(Arrays.toString(a));
        }
    }
}
```

Tema 0. Búsqueda y ordenación

Búsqueda

❑ Binaria o dicotómica

Para llevar a cabo una búsqueda binaria es requisito imprescindible que el array esté **ordenado**. Asumamos que está ordenado en orden creciente. La búsqueda binaria comienza comparando el valor **clave** (el que buscamos) con el valor que ocupa la posición central y pueden darse tres situaciones:

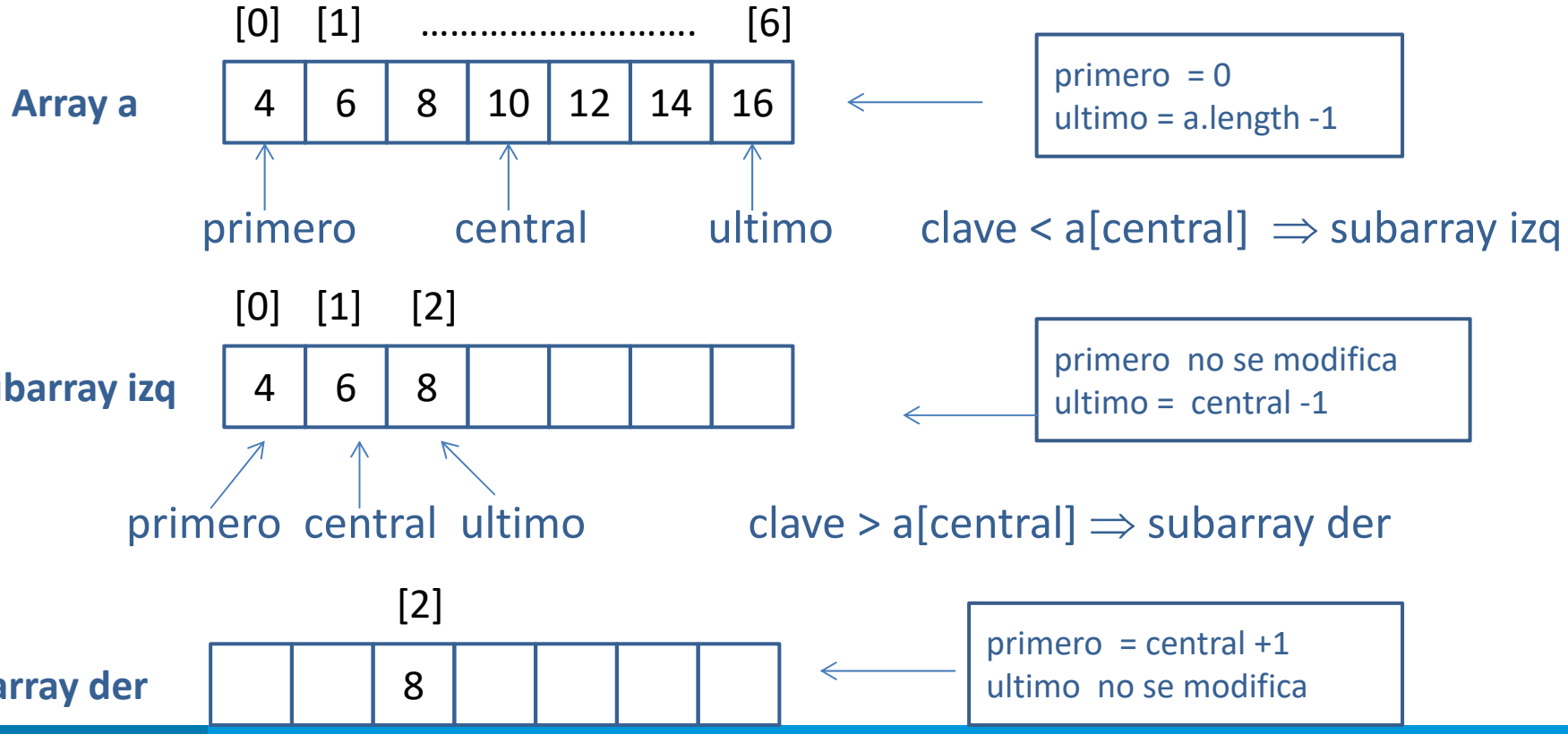
- La **clave** es **menor** que el valor del elemento que ocupa la posición central, podemos continuar la búsqueda en el subarray izquierdo.
- La **clave** es **mayor** que el valor del elemento que ocupa la posición central, podemos continuar la búsqueda en el subarray derecho.
- La **clave** es **igual** que el valor del elemento que ocupa la posición central, la búsqueda acaba.

Tema 0. Búsqueda y ordenación

Búsqueda

- **Binaria o dicotómica**

clave = 8



Tema 0. Búsqueda y ordenación

Ejemplo 2. Método de clase para la búsqueda binaria.

```
public static int busquedaBinaria(int [] a, int clave){    // a está ordenado
        int primero = 0;
        int ultimo = a.length-1;
        int central;
        if (clave < a[0] || clave > a[a.length -1])    //clave menor que el primer valor
            return -1;                                 // o mayor que el último valor
        else{
            while (ultimo >= primero){
                    central = (primero + ultimo)/ 2;
                    if (clave < a[central])        //subarray izquierdo
                            ultimo = central -1;
                    else if (clave > a[central])  //subarray derecho
                            primero = central + 1;
                    else
                    return central;
          }
        return -1;}}
```

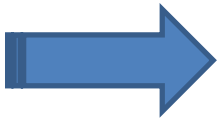
```java
package org.mp.tema00;

import java.util.Arrays;
import java.util.Scanner;

public class Busqueda2 {

    private static Scanner entrada;

    public static int busquedaBinaria(int [] a, int clave){
        int primero = 0;
        int ultimo = a.length-1;
        int central;
        if (clave < a[0] || clave > a[a.length -1])
            return -1;
        else{
            while (ultimo >= primero){
                central = (primero + ultimo)/ 2;
                if (clave < a[central]) //subarray izquierdo
                    ultimo = central -1;
                else if (clave > a[central])//subarray derecho
                    primero = central + 1;
                else
                    return central;
            }
            return -1;
        }
    }
```

```java

    public static void main(String[] args) {

        int[] a = { 4, 6, 8, 10, 12, 14, 16 };
        entrada = new Scanner(System.in);
        System.out.println("Introduce el valor que buscas en el array ");
        int clave = entrada.nextInt();
        int posicion = busquedaBinaria(a, clave);
        if (posicion == -1) {
            System.out.println("El valor " + clave + " no está en el array");
            System.out.println(Arrays.toString(a));
        } else {
            System.out.println("El valor " + clave
                    + " está en el array en la posición " + (posicion + 1));
            System.out.println(Arrays.toString(a));
        }
    }
}
```

Salida

```
Introduce el valor que buscas en el array
10
El valor 10 está en el array en la posición 4
[4, 6, 8, 10, 12, 14, 16]
```

Tema 0. Búsqueda y ordenación

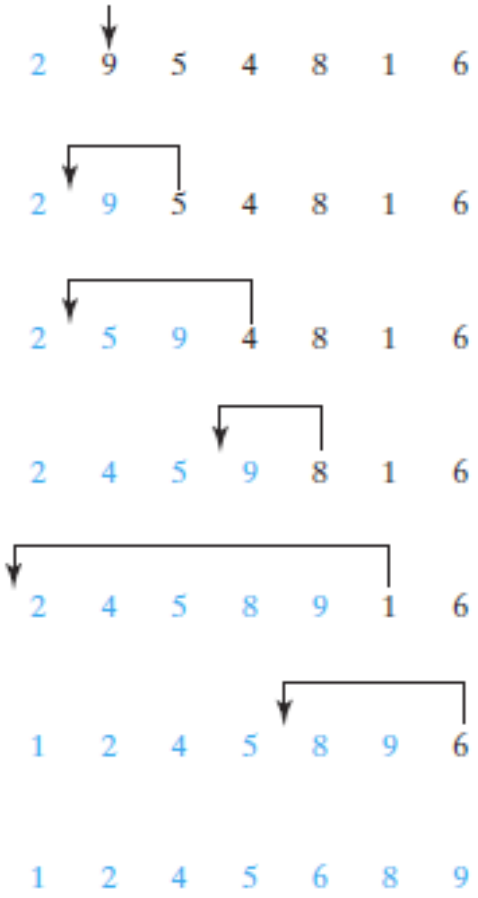


Ordenación

- Introducción
- Método de inserción
- Ordenación de un array de objetos

❑ Método de inserción

Ordena un array de valores considerando que cada vez inserta un nuevo elemento en una sublista ordenada.



Paso 1. Inicialmente la sublista ordenada contiene el primer elemento. Inserta 9 en la sublista.

Paso 2. La sublista ordenada es [2, 9]. Inserta 5 en dicha sublista.

Paso 3. La sublista ordenada es [2, 5, 9]. Inserta 4 en dicha sublista.

Paso 4. La sublista ordenada es [2, 4, 5, 9]. Inserta 8 en dicha sublista.

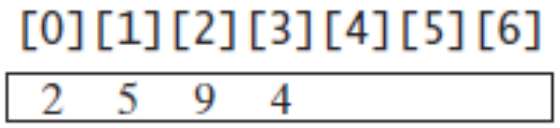
Paso 5. La sublista ordenada es [2, 4, 5, 8, 9]. Inserta 1 en dicha sublista.

Paso 6. La sublista ordenada es [1, 2, 4, 5, 8, 9]. Inserta 6 en dicha sublista.

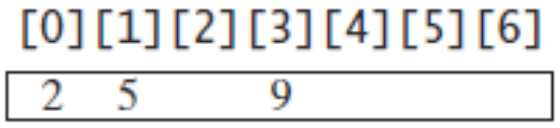
Paso 7. La lista entera está ordenada.

❑ Método de inserción

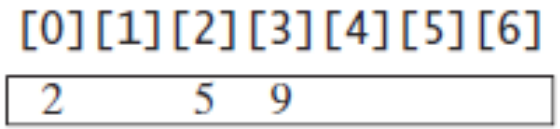
Supongamos que partimos de la sublista {2, 5, 9} y queremos incorporar 4.



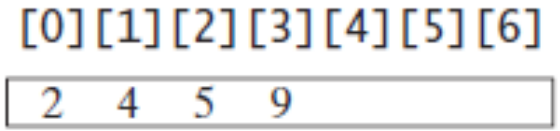
Paso 1. Guardamos 4 en una variable auxiliar.



Paso 2. Movemos a[2] a a[3] puesto que 4 < 9.



Paso 3. Movemos a[1] a a[2] puesto que 4 < 5.



Paso 4. Asignamos aux a a[1].

❑ Método de inserción

```
public static void insercion(int[] a) {
        for (int i = 1; i < a.length; i++) {
            int aux = a[i];
            int j;
            for (j = i - 1; j >= 0 && aux < a[j]; j--) {
                    a[j + 1] = a[j];
             }
            // Inserto el elemento (aux) en a[j + 1]
            a[j + 1] = aux;
        }
}
```

Bucle de las incorporaciones de cada elemento

Bucle de los desplazamientos

```java
package org.mp.tema00;

import java.util.Arrays;

public class Ordenacion {

    public static void insercion(int[] a) {
        for (int i = 1; i < a.length; i++) {
            int aux = a[i];
            int j;
            for (j = i - 1; j >= 0 && aux < a[j]; j--) {
                a[j + 1] = a[j];
            }
            // Inserto el elemento (aux) en a[j + 1]
            a[j + 1] = aux;
        }
    }

    public static void main(String[] args) {

        int[] a = { 2, 9, 5, 4, 8, 1, 6 };

        System.out.println("Array sin ordenar " + Arrays.toString(a));
        insercion(a);
        System.out.println("Array ordenado por el método de inserción " + Arrays.toString(a));

    }
```

Salida

```
Array sin ordenar [2, 9, 5, 4, 8, 1, 6]
Array ordenado por el método de inserción [1, 2, 4, 5, 6, 8, 9]
```

Tema 0. Búsqueda y ordenación

Ordenación

❑ Ordenación de un array de objetos

En este apartado se presenta un método de clase **genérico** para ordenar cualquier array de objetos siempre que dichos objetos sean instancias o ejemplares de clases que implementan la interface *Comparable* y que por tanto pueden utilizar el método *compareTo.*

Las clases *Integer, Double, Character* (todos los envoltorios) y *String* implementan la interface *Comparable*, esto significa que los objetos de estas clases pueden compararse utilizando el método *compareTo.* Asimismo la clase *Fraccion* que hemos diseñado también implementa la interface *Comparable.*

Vamos a reescribir los métodos de ordenación estudiados para que nos sirvan para ordenar cualquier array de objetos.

Método de selección

Ordenación array de enteros

```
public static void seleccion(int[] a) {
    for (int i = 0; i < a.length - 1; i++) {
        // Busco el mínimo en a[i ..a.length-1]
        int valorMinimo = a[i];
        int posicionMinimo = i;
        for (int j = i + 1; j < a.length; j++) {
            if (a[j] < valorMinimo) {
                valorMinimo = a[j];
                posicionMinimo = j;
            }
        }
        // Intercambio a[i] con a[posicionMinimo]
        //si es necesario
        if (posicionMinimo != i) {
            a[posicionMinimo] = a[i];
            a[i] = valorMinimo;
        }
    }
}
```

Ordenación array de Comparable

```
public static void seleccion(Comparable[] a) {
    for (int i = 0; i < a.length - 1; i++) {
        // Busco el mínimo en a[i ..a.length-1]
        Comparable valorMinimo = a[i];
        int posicionMinimo = i;
        for (int j = i + 1; j < a.length; j++) {
            if (a[j].compareTo(valorMinimo) < 0) {
                valorMinimo = a[j];
                posicionMinimo = j;
            }
        }
        // Intercambio a[i] con a[posicionMinimo]
        //si es necesario
        if (posicionMinimo != i) {
            a[posicionMinimo] = a[i];
            a[i] = valorMinimo;
        }
    }
}
```

¡MUCHAS GRACIAS!

UNIVERSIDAD DE ALMERÍA