

Búsqueda

Fernando Bobillo

Enunciado

- **Buscar** un elemento de una lista de números enteros
 - Devolver la **posición** del vector en la cual se encontró

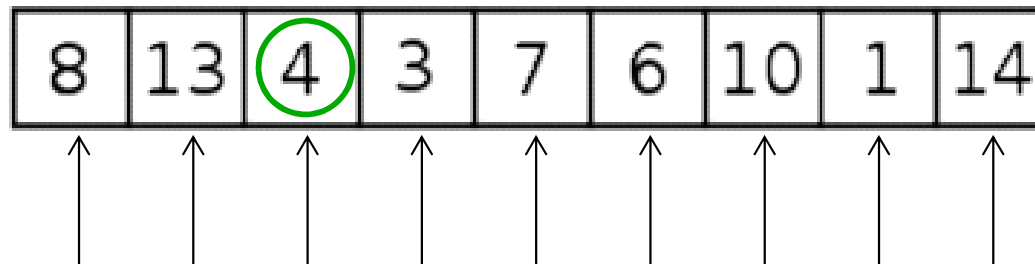


Variantes

- ¿Qué hacer si el elemento aparece **varias veces**?
 - Devolver la primera posición en que aparece
 - Devolver la última posición en que aparece
 - Devolver una de las posiciones aleatoriamente
 - Devolver una lista de todas las posiciones
- ¿Qué hacer si el elemento **no aparece**?
 - Devolver una posición inválida como -1
 - Escribir por pantalla un mensaje indicándolo
- Devolver si existe o no (valor **booleano**), pero la posición

Solución 1

- Recorrer exhaustivamente la lista: **búsqueda lineal**
- Inicialmente suponemos que **la posición es -1**
 - Este valor inválido indica que aún no se ha encontrado
- **Comparar cada elemento** con el elemento buscado
 - Si son iguales, guardar la posición en una variable
- **Al salir del bucle**, devolver la posición almacenada
 - Si hay varias apariciones, se devuelve la **última**
- Ejemplo: buscar el número 4

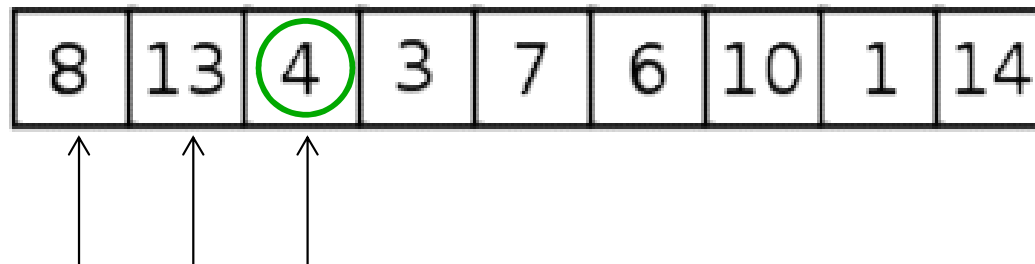


Código 1

```
public static int busqueda1(int[] a, int elementoBuscado) {  
    int posicion = -1;  
    for (int i = 0; i < a.length; i++)  
        if (a[i] == elementoBuscado)  
            posicion = i;  
    return posicion;  
}  
  
public static void main(String[] args) {  
    int[] a = {8, 13, 4, 3, 7, 6, 10, 1, 14};  
    System.out.println("Posición : " + busqueda1(a, 4));  
}
```

Solución 2

- Mejora: **salir del bucle** al encontrar el elemento
 - Si hay varias apariciones, se devuelve la **primera**
- Para ello, transformamos el bucle `for` en un bucle `while` que se repetirá mientras se cumplan **2 condiciones**
 1. Quedan posiciones del vector por **recorrer**
 - Es decir, i es menor que la longitud del vector
 2. El elemento buscado aún no se ha **encontrado**
 - Es decir, la posición es igual a -1
- Ejemplo: buscar el número 4



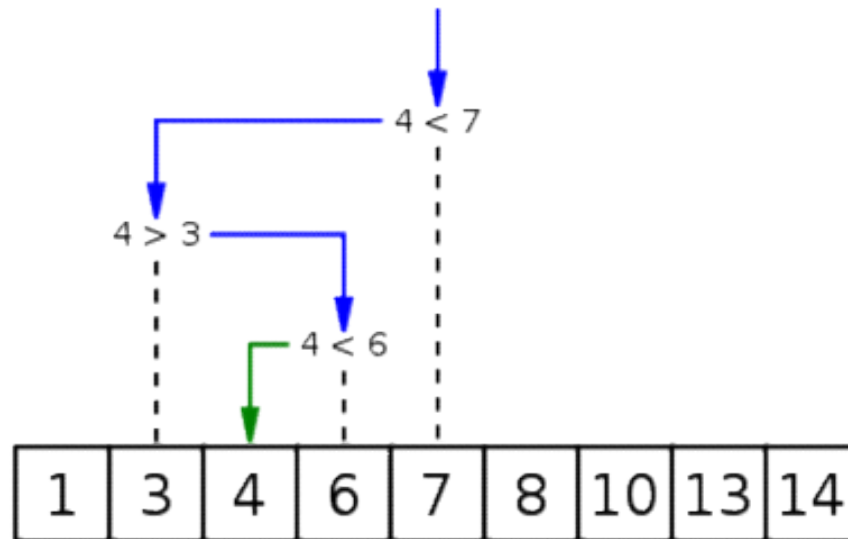
Código 2

```
public static int busqueda2(int[] a, int elementoBuscado) {  
    int posicion = -1;  
    int i = 0;  
    while ( (i < a.length) && (posicion == -1)) {  
        if (a[i] == elementoBuscado)  
            posicion = i;  
        i++;  
    }  
    return posicion;  
}
```

```
public static void main(String[] args) {  
    int[] a = {8, 13, 4, 3, 7, 6, 10, 1, 14};  
    System.out.println("Posición : " + busqueda2(a, 4));  
}
```

Solución 3

- Suposición: los elementos del vector están **ordenados**
- En cada paso, reducir a la mitad las posiciones en que buscar
 - Comparar con el elemento **central** del vector
 - Si el elemento buscado es menor, buscar en **mitad izquierda**
 - Si el elemento buscado es mayor, buscar en **mitad derecha**



Código 3

```
public static int busqueda3(int[] a, int elementoBuscado) {
    int izq = 0;
    int der = a.length - 1;
    while (izq <= der) {
        int centro = izq + (der - izq) / 2;
        if (elementoBuscado < a[centro])
            der = centro - 1;
        else if (elementoBuscado > a[centro])
            izq = centro + 1;
        else // elementoBuscado == a[centro]
            return centro;
    }
    return -1;
}

public static void main(String[] args) {
    int[] a = {1, 3, 4, 6, 7, 8, 10, 13, 14};
    System.out.println("Posición : " + busqueda3(a, 4));
}
```

Generación de números aleatorios

```
import java.util.*;

public class Busqueda {
    public static void main(String[] args) {
        Random r = new Random();
        Scanner sc = new Scanner(System.in);
        System.out.print("Dimensión de la lista de números: ");
        int size = sc.nextInt();
        int [] a = new int[size];
        for (int i=0; i<size; i++) {
            int x = r.nextInt();
            x = Math.abs(x) % 50;
            a[i] = x;
        }
        System.out.println("Posición : " + busqueda3(a, 4));
    }
}
```

La clase `Random` permite crear números aleatorios (necesita `java.util`)

Generación de números aleatorios

```
import java.util.*;
public class Busqueda {
    public static void main(String[] args) {
        Random r = new Random();
        Scanner sc = new Scanner(System.in);
        System.out.print("Dimensión de la lista de números: ");
        int size = sc.nextInt();
        int [] a = new int[size];
        for (int i=0; i<size; i++) {
            int x = r.nextInt();
            x = Math.abs(x) % 50;
            a[i] = x;
        }
        System.out.println("Posición : " + busqueda3(a, 4));
    }
}
```

Lee de teclado la dimensión del vector que queremos crear

Generación de números aleatorios

```
import java.util.*;
public class Busqueda {
    public static void main(String[] args) {
        Random r = new Random();
        Scanner sc = new Scanner(System.in);
        System.out.print("Dimensión de la lista de números: ");
        int size = sc.nextInt();
        int [] a = new int[size];
        for (int i=0; i<size; i++) {
            int x = r.nextInt();
            x = Math.abs(x) % 50;
            a[i] = x;
        }
        System.out.println("Posición : " + busqueda3(a, 4));
    }
}
```

Para cada casilla del vector, crearemos su valor aleatoriamente

Generación de números aleatorios

```
import java.util.*;
public class Busqueda {
    public static void main(String[] args) {
        Random r = new Random();
        Scanner sc = new Scanner(System.in);
        System.out.print("Dimensión de la lista de números: ");
        int size = sc.nextInt();
        int [] a = new int[size];
        for (int i=0; i<size; i++) {
            int x = r.nextInt();
            x = Math.abs(x) % 50;
            a[i] = x;
        }
        System.out.println("Posición : " + busqueda3(a, 4));
    }
}
```

Esto genera un número entero aleatorio (`nextDouble` obtiene un real)

Generación de números aleatorios

```
import java.util.*;
public class Busqueda {
    public static void main(String[] args) {
        Random r = new Random();
        Scanner sc = new Scanner(System.in);
        System.out.print("Dimensión de la lista de números: ");
        int size = sc.nextInt();
        int [] a = new int[size];
        for (int i=0; i<size; i++) {
            int x = r.nextInt();
            x = Math.abs(x) % 50;
            a[i] = x;
        }
        System.out.println("Posición : " + busqueda3(a, 4));
    }
}
```

Al calcular el resto de dividir entre n, nos aseguramos de que $x \in [0, n]$

Generación de números aleatorios

```
import java.util.*;
public class Busqueda {
    public static void main(String[] args) {
        Random r = new Random();
        Scanner sc = new Scanner(System.in);
        System.out.print("Dimensión de la lista de números: ");
        int size = sc.nextInt();
        int [] a = new int[size];
        for (int i=0; i<size; i++) {
            int x = r.nextInt();
            x = Math.abs(x) % 50;
            a[i] = x;
        }
        System.out.println("Posición : " + busqueda3(a, 4));
    }
}
```

Lo guardamos en la posición i-ésima del vector

Generación de números aleatorios

```
import java.util.*;
public class Busqueda {
    public static void main(String[] args) {
        Random r = new Random();
        Scanner sc = new Scanner(System.in);
        System.out.print("Dimensión de la lista de números: ");
        int size = sc.nextInt();
        int [] a = new int[size];
        for (int i=0; i<size; i++) {
            int x = r.nextInt();
            x = Math.abs(x) % 50;
            a[i] = x;
        }
        System.out.println("Posición : " + busqueda3(a, 4));
    }
}
```

Por último, llamamos al método de búsqueda e imprimimos el resultado

Código completo en Java



¡TU TURNO!