

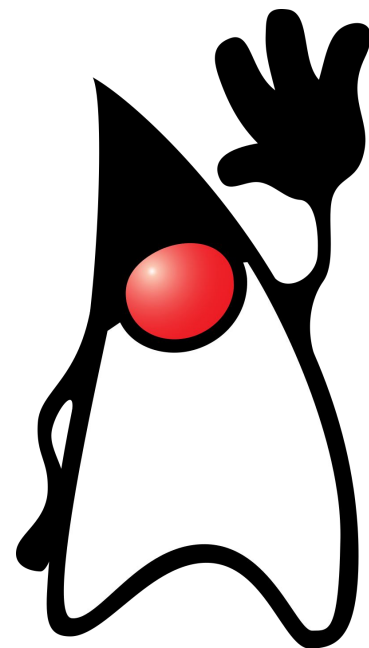


Collections - Arrays

Boris Molina Igor

Taller de programación II

bmolintai@ing.ucsc.cl



Imports

```
import java.util.Scanner;  
import java.util.List;  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.Arrays;
```


Collections

- Sort
- Reverse
- ReverseOrder
- BinarySearch
- Shuffle

API Java Collections: <https://docs.oracle.com/javase/8/docs/api/java/util/Collections.html>

BinarySearch

1	4	7	8	11	15	18	21	24	25	26	27	30	33	35	40
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15



$$\begin{aligned}\text{Promedio} &= (\text{min} + \text{max}) / 2 \\ &= (0 + 15) / 2 \\ &= 7.5 \approx 7\end{aligned}$$

BinarySearch


1	4	7	8	11	15	18	21	24	25	26	27	30	33	35	40
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Elemento a buscar : 18

Elemento central : 21

BinarySearch

1	4	7	8	11	15	18	21	24	25	26	27	30	33	35	40
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15



- Si es mayor, cambiar máximo a posición del elemento central - 1 (intervalo **azul**).
- Si es menor, cambiar mínimo a posición del elemento central + 1 (intervalo **rojo**).

BinarySearch

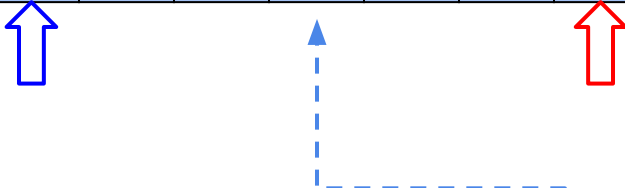
1	4	7	8	11	15	18	21	24	25	26	27	30	33	35	40
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15



Como 21 es mayor que 18, se repite tomando el intervalo de la **izquierda**.

BinarySearch

1	4	7	8	11	15	18	21	24	25	26	27	30	33	35	40
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15



$$\begin{aligned}\text{Promedio} &= (\text{min} + \text{max}) / 2 \\ &= (0 + 6) / 2 \\ &= 3\end{aligned}$$

BinarySearch



1	4	7	8	11	15	18	21	24	25	26	27	30	33	35	40
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Elemento a buscar : 18

Elemento central : 8

BinarySearch

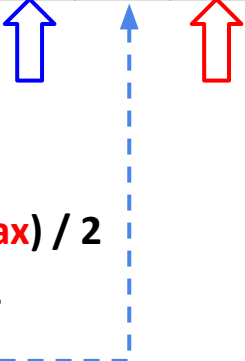
1	4	7	8	11	15	18	21	24	25	26	27	30	33	35	40
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Como 8 no es mayor que 18, se repite tomando el intervalo de la **derecha**.

BinarySearch

1	4	7	8	11	15	18	21	24	25	26	27	30	33	35	40
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15



$$\begin{aligned}\text{Promedio} &= (\text{min} + \text{max}) / 2 \\ &= (4 + 6) / 2 \\ &= 5\end{aligned}$$

BinarySearch



1	4	7	8	11	15	18	21	24	25	26	27	30	33	35	40
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Elemento a buscar : 18

Elemento central : 15

BinarySearch

1	4	7	8	11	15	18	21	24	25	26	27	30	33	35	40
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Como 15 no es mayor que 18, se repite tomando el intervalo de la **derecha**.

BinarySearch

1	4	7	8	11	15	18	21	24	25	26	27	30	33	35	40
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Elemento a buscar : 18

Elemento central : 18

Collections

- Sort
- Reverse
- ReverseOrder
- BinarySearch
- Shuffle

API Java Collections: <https://docs.oracle.com/javase/8/docs/api/java/util/Collections.html>

Collections métodos

```
List<Integer> arr = new ArrayList<>();
arr.add(7);
arr.add(1);
arr.add(4);
arr.add(11);
arr.add(8);
```

Tipo	Método	Descripción
void	<code>Collections.sort(List list);</code>	Ordena la lista especificada en orden ascendente, de acuerdo con el orden natural de sus elementos.
<code>Collections.sort(arr); //Ordena la lista ascendentemente [1, 4, 7, 8, 11]</code>		
Comparator	<code>Collections.reverseOrder();</code>	Devuelve un Comparator que impone el orden inverso al natural en una colección de objetos.
<code>Collections.reverseOrder() //Retorna un Comparator con el orden inverso al natural</code>		

Collections métodos

```
List<Integer> arr = new ArrayList<>();
arr.add(7);
arr.add(1);
arr.add(4);
arr.add(11);
arr.add(8);
```

Tipo	Método	Descripción
void	<pre>Collections.sort(List list, Comparator comp);</pre> <pre>Collections.sort(arr, Collections.reverseOrder());</pre> <pre>//Ordena la lista descendente [11, 8, 7, 4, 1]</pre>	<p>Ordena la lista especificada en orden descendente.</p>
void	<pre>Collections.reverse(List list);</pre> <pre>Collections.reverse(arr);</pre> <pre>//Invierte el orden de los elementos [8, 11, 4, 1, 7]</pre>	<p>Invierte el orden de los elementos en la lista especificada.</p>

Collections métodos

```
List<Integer> arr = new ArrayList<>();
arr.add(1);
arr.add(4);
arr.add(7);
arr.add(8);
arr.add(11);
```

Tipo	Método	Descripción
int	<code>Collections.binarySearch(List list, int number);</code>	Busca en la lista el objeto especificado mediante Búsqueda Binaria .
	<pre>int index = Collections.binarySearch(arr, 8); //Retorna el índice del número dado, en este caso retorna 3</pre>	
void	<code>Collections.shuffle(List list);</code>	Permuta aleatoriamente la lista especificada.
	<pre>Collections.shuffle(arr); //Desordena los elementos de la lista</pre>	

Arrays

- Sort
- BinarySearch
- ToString

API Java Arrays: <https://docs.oracle.com/javase/8/docs/api/java/util/Arrays.html>

Arrays métodos

```
int vec[] = new int[5];
vec[0] = 7;
vec[1] = 1;
vec[2] = 4;
vec[3] = 11;
vec[4] = 8;
```

Tipo	Método	Descripción
void	<code>Arrays.sort(int vec[]);</code>	Ordena el vector especificado en orden ascendente, de acuerdo con el orden natural de sus elementos.
	<code>Arrays.sort(vec); //Ordena el vector ascendentemente [1, 4, 7, 8, 11]</code>	
int	<code>Arrays.binarySearch(int vec[], int number);</code>	Busca en el vector el objeto especificado mediante Búsqueda Binaria .
	<code>int index = Arrays.binarySearch(vec, 8);</code> <code>//Retorna el índice del número dado, en este caso retorna 3</code>	

Arrays métodos

```
int vec[] = new int[5];  
vec[0] = 7;  
vec[1] = 1;  
vec[2] = 4;  
vec[3] = 11;  
vec[4] = 8;
```

Tipo	Método	Descripción
void	<code>Arrays.toString(int vec[]);</code>	Retorna una representación en String del vector especificado.
<pre>System.out.println(Arrays.toString(vec)); //Imprime el vector con el siguiente formato: [7, 1, 4, 11, 8]</pre>		

Competencia 2

Indicaciones Competencia 2 (Evaluada)

1. Entrar a vjudge.net.
2. Registrarse en la página con un “***Username***” y “***Nickname***” descriptivos.
 - a. En el campo ***School*** rellenen con “**UCSC**”
3. Luego entren a la siguiente competencia [Competencia 2](#).
4. **Password:** ucsc

Reglas Competencia 2 (Evaluada)

1. Con 0 problema resuelto = 10.
2. Con 1 problema resueltos = 40.
3. Con 2 problemas resueltos = 70.

Cómo subir una solución

1. Hacer clic en “***Submit***”.
2. En ***Problem***, seleccionar el problema del cual tiene la solución programada.
3. En ***Language***, seleccionar ***Java 1.8.0_241*** ó ***Java 1.8.0***.
4. En ***Share***, seleccionar ***NO***.
5. En ***Solution***,
 - a. Copiar su código SIN el package.
 - b. Incluir las bibliotecas (import).
 - c. El nombre de la clase principal debe de ser **Main**.

Ejemplo

Submit

Problem:

D - Problema D

Language:

Java 1.8.0_241

Share:

Yes

No

Solution:

```
import java.util.ArrayList;
import java.util.List;
import java.util.Collections;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        List<Integer> arr = new ArrayList<>();

        int n = sc.nextInt();

        for (int i = 0; i < n; i++) {
            arr.add(sc.nextInt());
        }

        Collections.sort(arr);

        for (int i = 0; i < arr.size(); i++) {
            System.out.print(arr.get(i) + " ");
        }
    }
}
```

Cancel

Submit



Collections - Arrays

Boris Molina Igor

Taller de programación II

bmolnai@ing.ucsc.cl

