**Ali Aauicha**

**Mi practica de Algoritmos**

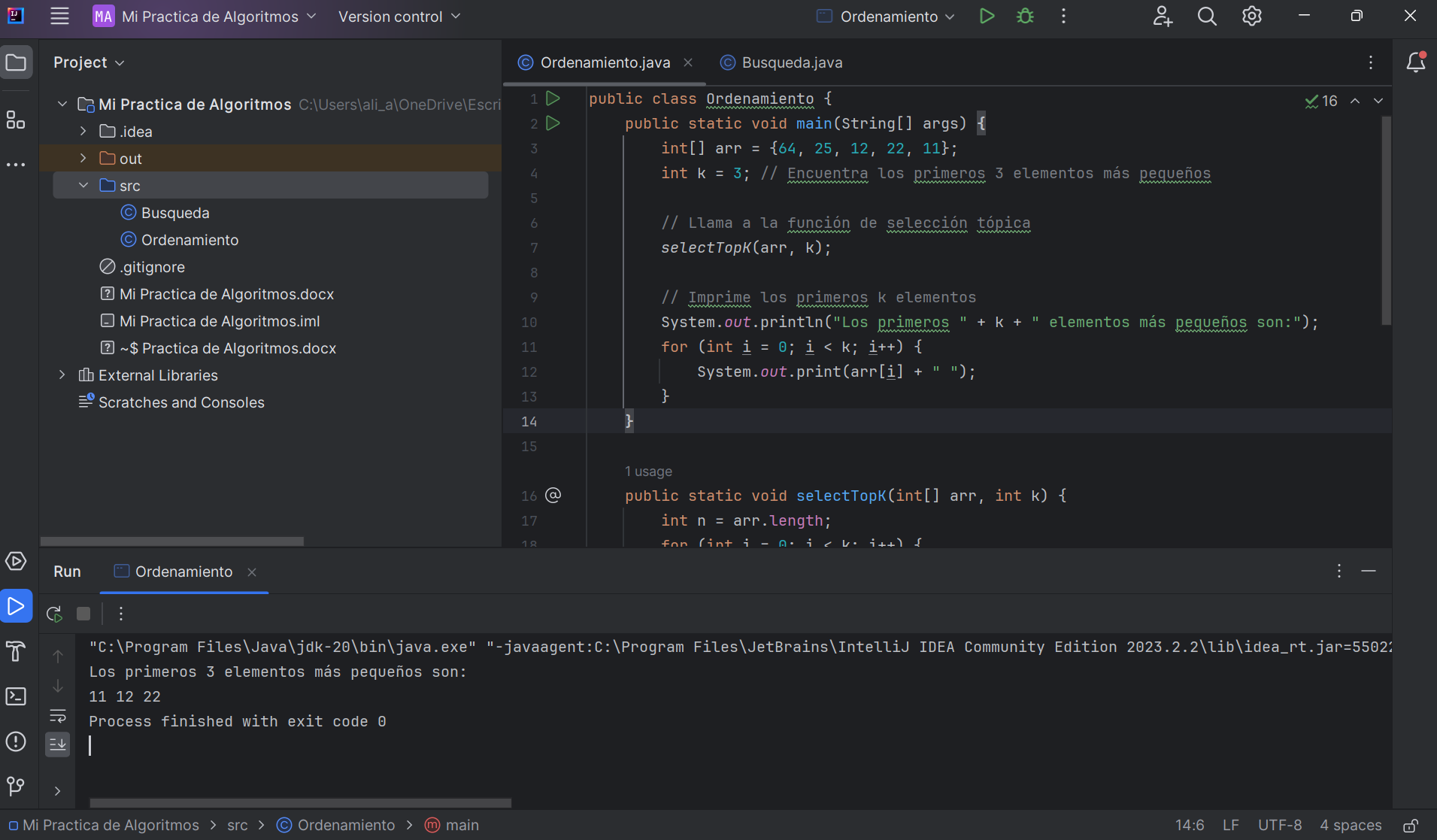
1. **Ordenamiento: Algoritmo de Selección Tópica (TopK Selection Algorithm):**

El algoritmo de selección tópica es utilizado para encontrar los primeros "k" elementos más pequeños (o más grandes) en una lista de elementos. En este ejemplo, ordenaremos un array de números en orden ascendente utilizando el algoritmo de selección tópica.

CODIGO JAVA:

public class Ordenamiento {  
 public static void main(String[] args) {  
 int[] arr = {64, 25, 12, 22, 11};  
 int k = 3; // Encuentra los primeros 3 elementos más pequeños  
  
 // Llama a la función de selección tópica  
 *selectTopK*(arr, k);  
  
 // Imprime los primeros k elementos  
 System.*out*.println("Los primeros " + k + " elementos más pequeños son:");  
 for (int i = 0; i < k; i++) {  
 System.*out*.print(arr[i] + " ");  
 }  
 }  
  
 public static void selectTopK(int[] arr, int k) {  
 int n = arr.length;  
 for (int i = 0; i < k; i++) {  
 int minIndex = i;  
 for (int j = i + 1; j < n; j++) {  
 if (arr[j] < arr[minIndex]) {  
 minIndex = j;  
 }  
 }  
 // Intercambia el elemento mínimo con el elemento en la posición actual  
 int temp = arr[i];  
 arr[i] = arr[minIndex];  
 arr[minIndex] = temp;  
 }  
 }  
}

RESULTADO:



1. **Búsqueda: Algoritmo de búsqueda de subcadenas**

El algoritmo de búsqueda de subcadenas se utiliza para encontrar la posición de una subcadena en una cadena más larga. En este ejemplo, buscaremos una subcadena en una cadena principal.

CODIGO DE JAVA:

public class Busqueda {  
 public static void main(String[] args) {  
 String cadenaPrincipal = "Esto es un ejemplo de búsqueda de subcadena.";  
 String subcadena = "ejemplo";  
  
 // Llama a la función de búsqueda de subcadena  
 int posicion = *buscarSubcadena*(cadenaPrincipal, subcadena);  
  
 if (posicion != -1) {  
 System.*out*.println("La subcadena se encuentra en la posición " + posicion + " de la cadena principal.");  
 } else {  
 System.*out*.println("La subcadena no se encuentra en la cadena principal.");  
 }  
 }  
  
 public static int buscarSubcadena(String cadenaPrincipal, String subcadena) {  
 int n = cadenaPrincipal.length();  
 int m = subcadena.length();  
  
 for (int i = 0; i <= n - m; i++) {  
 int j;  
 for (j = 0; j < m; j++) {  
 if (cadenaPrincipal.charAt(i + j) != subcadena.charAt(j)) {  
 break;  
 }  
 }  
 if (j == m) {  
 return i; // La subcadena se encontró en la posición i  
 }  
 }  
 return -1; // La subcadena no se encontró en la cadena principal  
 }  
}

RESULTADO:

