Lab #3: Algorithms

The main aim of the lab is to deal with some search algorithms.

==========

Task 1: Basic problems

==========

Task 1.1: Implement the following methods related to **linear search** in the class **MyArray.java**:

```
public class MyArray {
	private int[] array;

	public MyArray(int[] array) {
		this.array = array;
	}
// To find the index of the target in the array. If the target
// is not found in the array, then the method returns -1.
// Input: int[] array = {12, 10, 9, 45, 2, 10, 10, 45}, 45
// Output: 3

	public int iterativeLinearSearch(int target) {
		// TODO
		return 0;
	}

// To find the index of the target in the array. If the target
// is not found in the array, then the method returns -1.
// Input: int[] array = {12, 10, 9, 45, 2, 10, 10, 45}, 15
// Output: -1
	public int recursiveLinearSearch(int target) {
		// TODO
		return 0;
	}
```

==========

Task 1.2: Implement the following methods related to **binary search** in the class **MyArray.java**:

```
// To find the index of the target in the sorted array. If the
target is not found in the array, then the method returns -1.
	public int iterativeBinarySearch(int target) {
		// TODO
```

```
            return 0;
      }

// To find the index of the target in the sorted array. If the
target is not
// found in the array, then the method returns -1.
      public int recursiveBinarySearch(int target) {
            // TODO
            return 0;

      }
```

Task 1.3: How to change the implemented methods so that they can be used for the case in which the array is sorted by descending order.

==========

Task 2: Application of searching algorithms

==========

For a given **Product** and **OrderItem** classes as follows:

```
public class Product {
      private String id;
      private String name;
      private double price;
      private String type;

}

public class OrderItem {
      private Product p;
      private int quality;

}
```

Then, implement the following methods in **Order** class.

```
public class Order {
      private OrderItem[] items;

      public double cost() {
//          TODO
            return 0.0;
      }
```

```
    // using binary search approach
    public boolean contains(Product p) {
//        TODO
        return false;
    }

    // get all products based on the given type using
linear search
    public Product[] filter(String type) {
//        TODO
        return null;
    }

}
```

Suggestions:

- Consider using **Arrays.sort(T[] array)** or **Arrays.sort(T[] array, Comparator c)** to sort an array of objects.