

# Practical-05 Searching Algorithms

**Due** 13 Oct by 22:00      **Points** 100      **Available** after 11 Sep at 0:00

## Submission

```
1 | https://version-control.adelaide.edu.au/svn/<YOUR-ID>/2019/s2/fcs/week-08/practical-05
2 |
3 |
4 |
5 | save the projects you create in this practical.
6 | add to your svn repository and submit it to your web submission
```

## Searching Algorithms

**Question 1:** Set up a new project in your IDE with a Main class. Your project name should be `MySearchAlg` and it should be saved under practical-05.

**Question 2:** You need to build a base class for all your searching algorithms. You should name this base class as `MySearchAlg`. Since we cannot implement the search method in base class, you should declare it as an [abstract class](https://docs.oracle.com/javase/tutorial/java/land/abstract.html) [↗](https://docs.oracle.com/javase/tutorial/java/land/abstract.html) [\\_\(https://docs.oracle.com/javase/tutorial/java/land/abstract.html\)](https://docs.oracle.com/javase/tutorial/java/land/abstract.html)

```
1 | public abstract class MySearchAlg
2 |
3 |     Methods:
4 |     abstract int search(int[] array, int num);
```

**Question 3:** Define a **LinearSearch** class. This class should extend the `MySearchAlg` class. In this class, your search method should take a sequence of integer numbers and the number you want to find as input and return the index of that number. The index starts from 0, return -1 when it does not exist.

```
public class LinearSearch
Methods:
public int search(int[] array, int num);
```

**Question 4:** Define a **BinarySearch** class. This class should extend the `MySearchAlg` class. In this class, your search method should take a sequence of integer numbers and the number you want to find as input and return the index of that number. The index starts from 0, return -1 when it does not exist.

```
public class BinarySearch
Methods:
public int search(int[] array, int num);
```

**Question 5:** In your **Main** class, build a test method to test your code.

```
private static boolean test(int result, int ans);
```

The test method should take two integers as input, the result is an integer returned by one of your searching algorithms, and the ans is an integer generated by yourself. Your test method should compare if these two integers are the same, return true when they are the same and false when they not. You should generate at least 3 test cases and use these test cases to test all searching algorithms above.

**Question 6:** In your Main class, build a findMaxVal method.

```
private static int findMaxVal(int array[]);
```

This method takes an array of elements which is first increasing and then decreasing as input. And return the index value for the maximum number in this array. The same element will not appear in one array twice. For example,

Input: array = {1,5,8,12,9,7,-1}

Output: 3

Input: array = {1,15,0}

Output: 1