DAY 4

ALGORITHMS

Linear Search - Tryout

```
class Tester {

	public static int search(int arrayOfElements[], int elementToBeSearched) {
		for (int index = 0; index < arrayOfElements.length; index++) {
			// Returning the index of the element if the element is found
			if (arrayOfElements[index] == elementToBeSearched)
				return index;
		}
		// Returning -1 if the element is not found
		return -1;
	}

	public static void main(String[] args) {
		int[] arrayOfElements = { 10, 39, 45, 47, 50, 15, 23, 32, 25, 49 };
		int elementToBeSearched = 50;

		int index = search(arrayOfElements, elementToBeSearched);

		// Checking whether the element has been found or not
		if (index == -1)
			System.out.println("Element is not present in the array!");
		else
			System.out.println("Element is found at index position " + index+"!");
	}
}
```

Linear Search - Exercise 1

```
class Tester {

   public static int searchCustomerId(int[] customerIds, int customerIdToBeSearched) {
      int left = 0;
      int right = customerIds.length - 1;

      while (left <= right) {
```

```
        int mid = left + (right - left) / 2;

        if (customerIds[mid] == customerIdToBeSearched) {
            return mid; // Customer ID found at index mid
        }

        if (customerIds[mid] < customerIdToBeSearched) {
            left = mid + 1; // Search in the right half
        } else {
            right = mid - 1; // Search in the left half
        }
      }

      return -1; // Customer ID not found
   }

   public static void main(String[] args) {
      int[] customerIds = { 80451, 80462, 80465, 80479, 80550, 80561, 80665, 80770 };
      int customerIdToBeSearched = 80462;

      int index = searchCustomerId(customerIds, customerIdToBeSearched);

      if (index == -1)
         System.out.println("Customer Id " + customerIdToBeSearched + " is not found!");
      else
         System.out.println("Customer Id " + customerIdToBeSearched + " is found at index
position " + index + "!");
   }
}
```

Binary Search - Tryout 1

```
class Tester {

        public static int search(int arrayOfElements[], int elementToBeSearched) {
                int low = 0;
                int high = arrayOfElements.length - 1;
                int mid;
                while (low <= high) {
                        mid = (low + high) / 2;

                        // Checking if the element to be searched is present in the mid position
                        if (arrayOfElements[mid] == elementToBeSearched)
                                return mid;

                        // If the element to be searched is greater than the element in the mid
position, low is updated
```

```
                if (arrayOfElements[mid] < elementToBeSearched)
                        low = mid + 1;

                // If the element to be searched is lesser than the element in the mid
position, high is updated
                else
                        high = mid - 1;
        }

        // -1 is being returned when the element is not present in the array
        return -1;
    }

    public static void main(String[] args) {
        int[] arrayOfElements = { 10, 15, 23, 25, 32, 39, 45, 47, 49, 50 };
        int elementToBeSearched = 50;

        int index = search(arrayOfElements, elementToBeSearched);
        if (index == -1)
            System.out.println("Element is not present in the array!");
        else
            System.out.println("Element is found at index position " + index+"!");
    }
}
```

Binary Search - Tryout 2

```
class Tester {

    public static int search(int arrayOfElements[], int low, int high, int
elementToBeSearched) {
        if (low <= high) {
            int mid = (low + high) / 2;

            if (arrayOfElements[mid] == elementToBeSearched)
                return mid;

            if (arrayOfElements[mid] < elementToBeSearched)
                return search(arrayOfElements, mid + 1, high,
elementToBeSearched);

            return search(arrayOfElements, low, mid - 1, elementToBeSearched);
        }
        return -1;
```

```
	}

	public static void main(String[] args) {
		int[] arrayOfElements = { 10, 15, 23, 25, 32, 39, 45, 47, 49, 50 };
		int elementToBeSearched = 50;
		int low = 0;
		int high = arrayOfElements.length - 1;
		int index = search(arrayOfElements, low, high, elementToBeSearched);
		if (index == -1)
			System.out.println("Element is not present in the array!");
		else
			System.out.println("Element is found at index position " + index+"!");
	}
}
```

Binary Search - Exercise 1

```
class Tester {

   public static int iterations = 0;

   public static int searchElement(int[] elements, int low, int high, int elementToBeSearched) {
      while (low <= high) {
         iterations++;
         int mid = low + (high - low) / 2;

         if (elements[mid] == elementToBeSearched) {
            return mid; // Element found at index mid
         }

         if (elements[mid] < elementToBeSearched) {
            low = mid + 1; // Search in the right half
         } else {
            high = mid - 1; // Search in the left half
         }
      }

      return -1; // Element not found
   }

   public static void main(String[] args) {
      int[] elements = { 1, 23, 43, 46, 78, 90 };
      int elementToBeSearched = 43;

      int indexPosition = searchElement(elements, 0, elements.length - 1, elementToBeSearched);

      if (indexPosition == -1) {
```

```
            System.out.println("Element not found!");
        } else {
            System.out.println("Element found at index position " + indexPosition + "!");
        }

        System.out.println("Number of iterations: " + iterations);
    }
}
```