

Linear Searching

Time Complexity :

- Best Case : $O(1)$
- Worst Case : $O(n)$ [where n is the size of array]
- Q. Integer in Array

```
package com.inclass;

public class Integer {

    public static void main(String[] args) {
        int[] nums = {23, 45, 1, 2, 8, 19, -3, 16, -11, 28};
        int target = 2;
        System.out.println(linearSearch(nums, target));
        System.out.println(linearSearch2(nums, target));
        System.out.println(rangeSearch(nums, target, 1, 5));
        System.out.println(findMin(nums));
    }

    // search in the array: return the index if item found
    // otherwise if item not found return -1
    static int linearSearch(int[] arr, int target) {
        if (arr.length == 0) {
            return -1;
        }
        // run a for loop
        for (int index = 0; index < arr.length; index++) {
            if (arr[index] == target) {
                return index;
            }
        }
        // no target found
        return -1;
    }

    // search in the array: return the element
    static int linearSearch2(int[] arr, int target) {
        if (arr.length == 0) {
            return java.lang.Integer.MAX_VALUE;
        }
        // run a for loop
        for (int element : arr) {
            if (element == target) {
                return element;
            }
        }
        // not target found
        return java.lang.Integer.MAX_VALUE;
    }

    // search in array[index1, index2]: return true or false
    static int rangeSearch(int[] arr, int target, int index1, int index2) {
        if (index1 >= index2) {
```

```

        return -1;
    }
    for (int i = index1; i < index2; i++) {
        if (arr[i] == target) {
            return i;
        }
    }
    return -1;
}
}

```

– Q. Searching Strings in Array

```

package com.inclass;

import java.util.Arrays;

public class Strings {

    public static void main(String[] args) {
        String name = "Driptanil";
        char target = 'a';
        System.out.println(search(name, target));
        System.out.println(Arrays.toString(name.toCharArray()));
    }
    static boolean search(String str, char target) {
        if (str.length() == 0) {
            return false;
        }
        // for (int i = 0; i < str.length(); i++) {
        //     if (str.charAt(i) == target) {
        //         return true;
        //     }
        // }
        for (char ch: str.toCharArray()) {
            if (ch == target) {
                return true;
            }
        }
        return false;
    }
}

```

– Q. Search Integer in Array

```

package com.inclass;

import java.util.Arrays;

public class Search2D {

    public static void main(String[] args) {
        int [][] arr ={
            {23, 4, 1},
            {18, 12, 3, 9},
            {78, 99, 34, 56},
            {18, 12}
        };
    }
}

```

```

        int target = 34;
        System.out.println(Arrays.toString(search(arr, target)));
    }

    // search target in array: return index
    static int[] search(int[][] arr, int target) {
        if (arr.length == 0) {
            return new int[] {-1, -1};
        }
        for (int row = 0; row < arr.length; row++) {
            for (int col = 0; col < arr[row].length; col++) {
                if (arr[row][col] == target){
                    return new int[] {row, col};
                }
            }
        }
        return new int[] {-1, -1};
    }
}

```

– Q. Search in 2-D Array

```

package com.inclass;

import java.util.Arrays;

public class Search2D {

    public static void main(String[] args) {
        int [][] arr ={
            {23, 4, 1},
            {18, 12, 3, 9},
            {78, 99, 34, 56},
            {18, 12}
        };
        int target = 34;
        System.out.println(Arrays.toString(search(arr, target)));
    }

    // search target in array: return index
    static int[] search(int[][] arr, int target) {
        if (arr.length == 0) {
            return new int[] {-1, -1};
        }
        for (int row = 0; row < arr.length; row++) {
            for (int col = 0; col < arr[row].length; col++) {
                if (arr[row][col] == target){
                    return new int[] {row, col};
                }
            }
        }
        return new int[] {-1, -1};
    }
}

```