

Exercise01

```
// O(n) time complexity

public static int searchForTarget(int[] arr , int target) {
    if (arr == null || arr.length == 0) {
        return -1;
    }
    for (int i = 0; i < arr.length; i++) {
        if (arr[i] == target) {
            return i;
        }
    }
    return -1;
}
```

// and the Output gonna be:

```
"C:\Program Files\Java\jdk-21\bin\java.exe" --enable-preview "-javaagent:C:\IntelliJ IDEA 2024.3.1.1\lib\idea_rt.jar=64886:C:\IntelliJ IDEA 2024.3.1.1\bin" -Dfile.encoding=UTF-8
When the function O(n) time complexity running the result is:
array is: [0, 2, 4, 6, 8]
The location of -1 is: -1 [Not Found]
The location of 0 is: 0 [Found at index: 0]
The location of 1 is: -1 [Not Found]
The location of 2 is: 1 [Found at index: 1]
The location of 3 is: -1 [Not Found]
The location of 4 is: 2 [Found at index: 2]
The location of 5 is: -1 [Not Found]
The location of 6 is: 3 [Found at index: 3]
The location of 7 is: -1 [Not Found]
The location of 8 is: 4 [Found at index: 4]
The location of 9 is: -1 [Not Found]

Process finished with exit code 0
```

```
// O(log n) time complexity

public static int searchForTarget(int[] arr, int target) {
    if (arr == null || arr.length == 0) {
        return -1;
    }
    int low = 0;
    int high = arr.length - 1;
    while (low <= high) {
        int mid = low + (high - low) / 2;
        if (arr[mid] == target) {
            return mid;
        }
        if (arr[mid] < target) {
            low = mid + 1;
        }
        else {
            high = mid - 1;
        }
    }
    return -1;
}

// and the Output gonna be:
```

```
"C:\Program Files\Java\jdk-21\bin\java.exe" --enable-preview "-javaagent:C:\IntelliJ IDEA 2024.3.1.1\lib\idea_rt.jar=65056:C:\IntelliJ IDEA 2024.3.1.1\bin" -Dfile.encoding=UTF-8
When the function O(log n) time complexity running the result is:
array is: [0, 2, 4, 6, 8]
The location of -1 is: -1 [Not Found]
The location of 0 is: 0 [Found at index: 0]
The location of 1 is: -1 [Not Found]
The location of 2 is: 1 [Found at index: 1]
The location of 3 is: -1 [Not Found]
The location of 4 is: 2 [Found at index: 2]
The location of 5 is: -1 [Not Found]
The location of 6 is: 3 [Found at index: 3]
The location of 7 is: -1 [Not Found]
The location of 8 is: 4 [Found at index: 4]
The location of 9 is: -1 [Not Found]

Process finished with exit code 0
```

```
//So, The output for first function is matching with second one.
```

Exercise02

```
// O(n) time complexity

public static int fibonacci(int n) {
    if (n < 0) {
        return -1;
    } else if (n == 0) {
        return 0;
    } else {
        int[] fibonacci = new int[n + 1];
        fibonacci[0] = 0;
        fibonacci[1] = 1;
        for (int i = 2; i <= n; i++) {
            fibonacci[i] = fibonacci[i - 1] + fibonacci[i - 2];
        }
        return fibonacci[n];
    }
}

// and the Output gonna be:
```

```
"C:\Program Files\Java\jdk-21\bin\java.exe" --enable-preview "-javaagent:C:\IntelliJ IDEA 2024.3.1.1\lib\idea_rt.jar=65467:C:\IntelliJ IDEA 2024.3.1.1\bin" -Dfile.encoding=UTF-8
When the function O(n) time complexity running the result is:
When the input is: -5 The result is: -1 Invalid Input
When the input is: 0 The result is: 0
When the input is: 5 The result is: 5
When the input is: 10 The result is: 55
When the input is: 15 The result is: 610
When the input is: 20 The result is: 6765
When the input is: 25 The result is: 75025
When the input is: 30 The result is: 832040
```

```
//O(2^n) time complexity

public static long fibonacci(int n) {
    if (n < 0) {
        return -1;
    }
    if (n == 0) {
        return 0;
    }
    if (n == 1) {
        return 1;
    }
    return fibonacci(n - 1) + fibonacci(n - 2);
}

// and the Output gonna be:
```

```
"C:\Program Files\Java\jdk-21\bin\java.exe" --enable-preview "-javaagent:C:\IntelliJ IDEA 2024.3.1.1\lib\idea_rt.jar=49440:C:\IntelliJ IDEA 2024.3.1.1\bin" -Dfile.encoding=UTF-8
When the function O(2^n) time complexity running the result is:
When the input is: -5 The result is: -1 Invalid Input
When the input is: 0 The result is: 0
When the input is: 5 The result is: 5
When the input is: 10 The result is: 55
When the input is: 15 The result is: 610
When the input is: 20 The result is: 6765
When the input is: 25 The result is: 75025
When the input is: 30 The result is: 832040
```

```
//So, The output for first function is matching with second one.
```

Exercise03

```
// O(n) time complexity
// O(1) space complexity

public static long factorial(int n) {
    if (n < 0){
        return -1;
    }
    int number = n;
    long factorial = 1;
    for (int i = 1; i <= number; i++) {
        factorial *= i;
    }
    return factorial;
}
// and the Output gonna be:
```

```
"C:\Program Files\Java\jdk-21\bin\java.exe" --enable-preview "-javaagent:C:\IntelliJ IDEA 2024.3.1.1\lib\idea_rt.jar=49602:C:\IntelliJ IDEA 2024.3.1.1\bin" -Dfile.encoding=UTF-8
when the function that has O(n) time complexity and O(1) space complexity running the result is:
The factorial of -5 is -1 Invalid Input
The factorial of 0 is 1
The factorial of 5 is 120
The factorial of 10 is 3628800
The factorial of 15 is 1307674368000
The factorial of 20 is 2432902008176640000
The factorial of 25 is 7034535277573963776
The factorial of 30 is -8764578968847253504
```

```
// O(n) time complexity
// O(n) space complexity

public static long factorial(int n) {
    if (n < 0){
        return -1;
    }
    if (n == 0){
        return 1;
    }
    long[] factorial = new long[n + 1];
    factorial[0] = 1;
    factorial[1] = 1;
    for (int i = 2; i <= n; i++){
        factorial[i] = factorial[i - 1] * i;
    }
    return factorial[n];
}
// and the Output gonna be:
```

```
"C:\Program Files\Java\jdk-21\bin\java.exe" --enable-preview "-javaagent:C:\IntelliJ IDEA 2024.3.1.1\lib\idea_rt.jar=49655:C:\IntelliJ IDEA 2024.3.1.1\bin" -Dfile.encoding=UTF-8
when the function that has O(n) time complexity and O(n) space complexity running the result is:
The factorial of -5 is -1 Invalid Input
The factorial of 0 is 1
The factorial of 5 is 120
The factorial of 10 is 3628800
The factorial of 15 is 1307674368000
The factorial of 20 is 2432902008176640000
The factorial of 25 is 7034535277573963776
The factorial of 30 is -8764578968847253504
```

```
//So, The output for first function is matching with second one.
```