Course title : **CSE2001**

Course title : **Data Structures and Algorithms**

Module : **5**

Topic : **1**

Introduction to Searching Algorithms

Objectives

This session will give the knowledge about

- Introduction to Searching Algorithms
- Linear Search
- Binary Search

Introduction to Sorting Algorithms

Search is a process of finding a value in a list of values. In other words, searching is the process of locating given value position in a list of values.

We can distinguish two types of searching.

- Linear Search Algorithm (Sequential Search Algorithm)
- Binary Search Algorithm

Linear Search Algorithm

Linear search is implemented using following steps.

Step 1 - Read the search element from left to right.

Step 2 - Compare the search element with every element in the list.

Step 3 - If found, return the position, else return -1

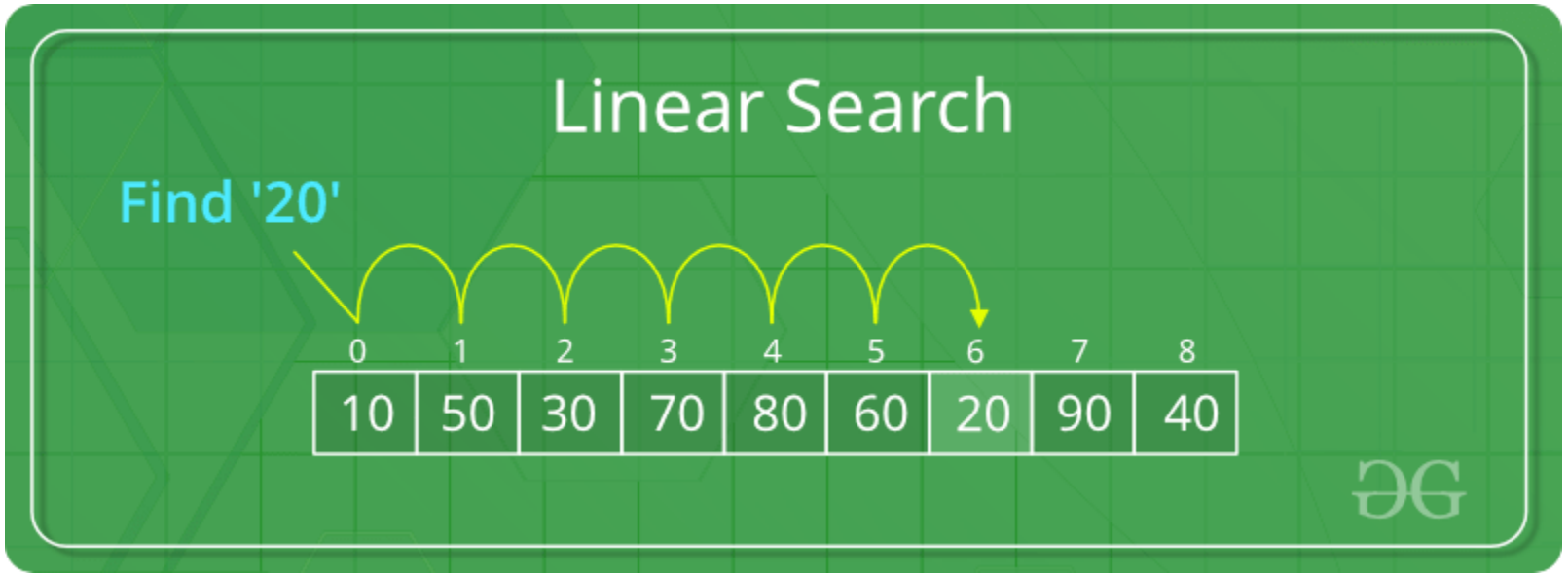
Features of Linear Search Algorithm

It is used for unsorted and unordered small list of elements.

It has a time complexity of O(n), which is not bad, but not that good too.

It has a very simple implementation.

Linear Search Algorithm



Linear Search Algorithm

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

Binary Search Algorithm

Binary search is a fast search algorithm with run-time complexity of O(log n).

This search algorithm works on the principle of divide and conquer.

For this algorithm to work properly, the data collection should be in the sorted form.

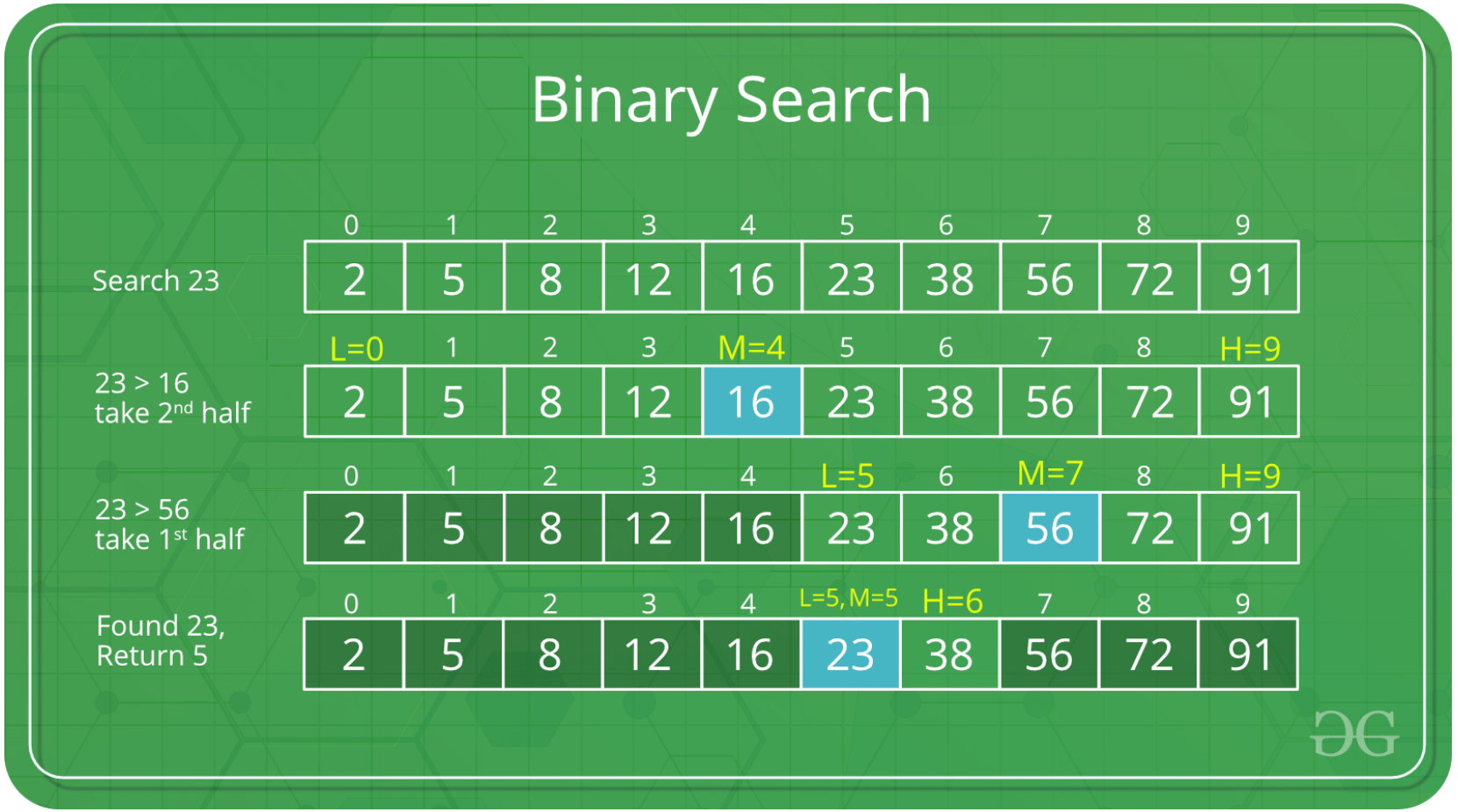
Features of Binary Search

It is great to search through large sorted arrays.

It has a time complexity of O(log n) which is a very good time complexity.

It has a simple implementation.

Binary Search Algorithm



Binary Search Algorithm

```
public static int binarySearch(int arr[], int first, int last, int key) {
    int mid = (first + last) / 2;
    while (first <= last) {
        if (arr[mid] < key)
            first = mid + 1;
        else if (arr[mid] == key)
            return mid;
        else
            last = mid - 1;
        mid = (first + last) / 2;
    }
    if (first > last)
        return -1;
    return mid;
}
```

Summary

At the end of this session we learned about

- Searching Algorithms
- Linear Search
- Binary Search