```
        return -1;

}

int main(){

        // Sorted array for binary search
        // int arr[] = {1,3,5,56,64,72,83,86,90,105,201};
        // int size = sizeof(arr)/sizeof(int);
        // int element = 225;
        // int searchIndex = binarySearch(arr, size, element);
        // printf("The element %d was found at index %d \n", element, searchIndex);
        return 0;
}
```

Searching:

Searching in data structure refers to the process of finding location LOC of an element in a list. This is one of the important parts of many data structures algorithms, as one operation can be performed on an element if and only if we find it. Various algorithms have been defined to find whether an element is present in the collection of items or not. This algorithm can be executed on both internal as well as external data structures. The efficiency of searching an element increases the efficiency of any algorithm.

Searching Algorithms are designed to check for an element or retrieve an element from any data structure where it is stored.

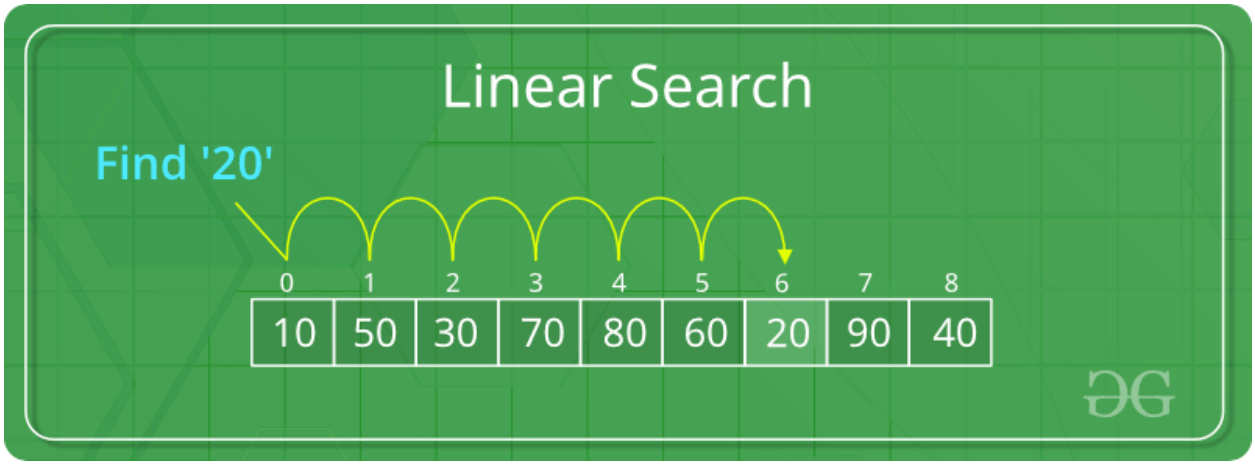
Searching in data-strucutre refers to the process of finding a desired element in set of items. The desired element is called "target". The set of items to be searched in, can be any data-structure like − list, array, linked-list, tree or graph.

Search refers to locating a desired element of specified properties in a collection of items.

Based on the type of search operation, these algorithms are generally classified into two categories:

Sequential Search: In this, the list or array is traversed sequentially and every element is checked.

Linear Search to find the element “20” in a given list of numbers



Linear-Search

Advantages of Linear Search:

- Linear search can be used irrespective of whether the array is sorted or not. It can be used on arrays of any data type.
- Does not require any additional memory.
- It is a well-suited algorithm for small datasets.

Drawbacks of Linear Search:

- Linear search has a time complexity of O(N), which in turn makes it slow for large datasets.
- Not suitable for large arrays.

When to use Linear Search?

- When we are dealing with a small dataset.
- When you are searching for a dataset stored in contiguous memory.

How Does Linear Search Algorithm Work?

In Linear Search Algorithm,

- Every element is considered as a potential match for the key and checked for the same.
- If any element is found equal to the key, the search is successful and the index of that element is returned.

- If no element is found equal to the key, the search yields “No match found”.

Interval Search: These algorithms are specifically designed for searching in sorted data-structures. These type of searching algorithms are much more efficient than Linear Search as they repeatedly target the center of the search structure and divide the search space in half. For Example: Binary Search.

Conditions for when to apply Binary Search in a Data Structure:

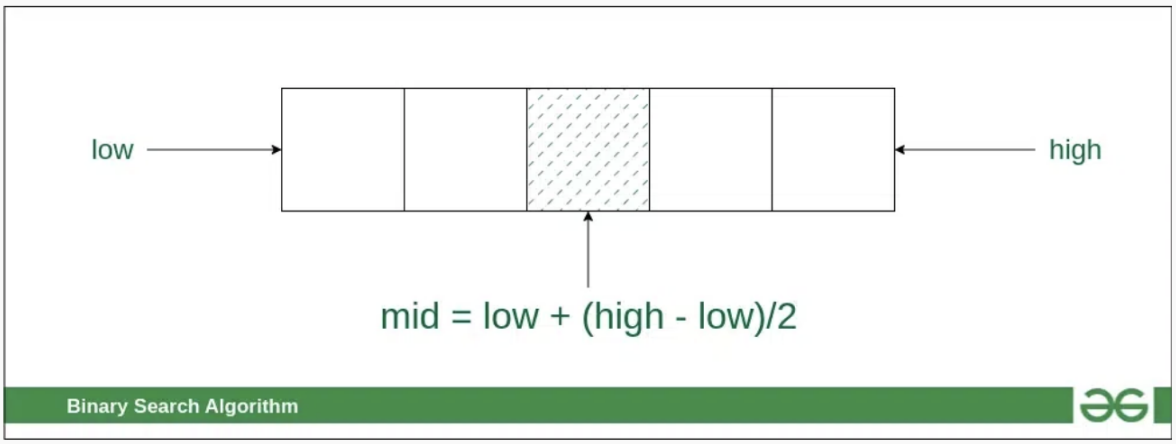
To apply Binary Search algorithm:

- The data structure must be sorted.
- Access to any element of the data structure takes constant time.

Binary Search Algorithm:

In this algorithm,

- Divide the search space into two halves by **finding the middle index “mid”**.

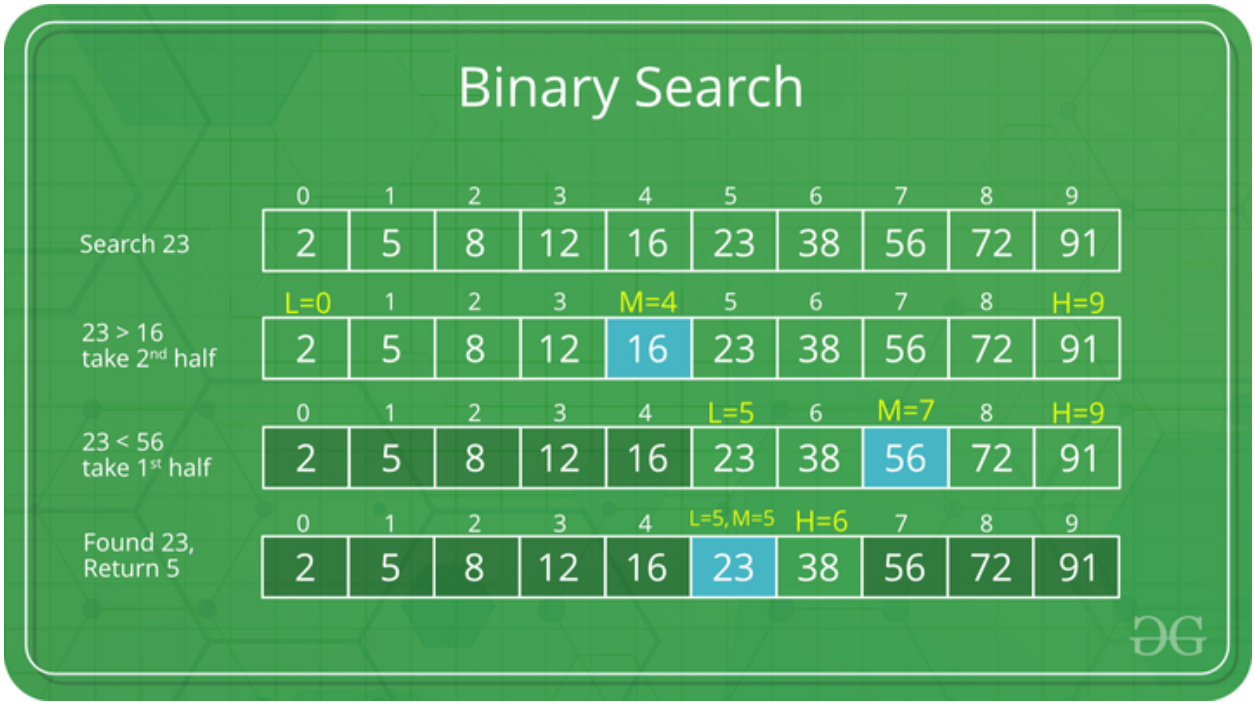


Finding the middle index “mid” in Binary Search Algorithm

Finding the middle index “mid” in Binary Search Algorithm

- Compare the middle element of the search space with the key.
- If the key is found at the middle element, the process is terminated.
- If the key is not found at the middle element, choose which half will be used as the next search space.
    - If the key is smaller than the middle element, then the left side is used for the next search.
    - If the key is larger than the middle element, then the right side is used for the next search.
- This process is continued until the key is found or the total search space is exhausted.

Binary Search to find the element “23” in a given list of numbers



Binary Search

How to Implement Binary Search?

The **Binary Search Algorithm** can be implemented in the following two ways

- Iterative Binary Search Algorithm
- Recursive Binary Search Algorithm

Linear Search	Binary Search
In linear search input data need not to be in sorted.	In binary search input data need to be in sorted order.
It is also called sequential search.	It is also called half-interval search.
The time complexity of linear search **O(n)**.	The time complexity of binary search **O(log n)**.
Multidimensional array can be used.	Only single dimensional array is used.
Linear search performs equality comparisons	Binary search performs ordering

	comparisons
It is less complex.	It is more complex.
It is very slow process.	It is very fast process.

Advantages of Binary Search:

- Binary search is faster than linear search, especially for large arrays.
- More efficient than other searching algorithms with a similar time complexity, such as interpolation search or exponential search.
- Binary search is well-suited for searching large datasets that are stored in external memory, such as on a hard drive or in the cloud.

Drawbacks of Binary Search:

- The array should be sorted.
- Binary search requires that the data structure being searched be stored in contiguous memory locations.
- Binary search requires that the elements of the array be comparable, meaning that they must be able to be ordered.

Applications of Binary Search:

- Binary search can be used as a building block for more complex algorithms used in machine learning, such as algorithms for training neural networks or finding the optimal hyperparameters for a model.
- It can be used for searching in computer graphics such as algorithms for ray tracing or texture mapping.
- It can be used for searching a database.