Searching Algorithms

by Hasan Murad

Searching is one of the most basic operations and efficient algorithm for searching is crying need for different fields of computer science. In this Chapter, we are going to discuss on various searching algorithms in incremental approach by starting with simple algorithms first. Moreover, we evaluate each algorithm with runtime and space complexity analysis and represent a comparison among all existing algorithms on searching.

Classification of Searching Algorithms:

Searching Algorithms are designed to check for an element or retrieve an element from any data structure where it is stored. Based on the type of search operation, these algorithms are generally classified into two categories:

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

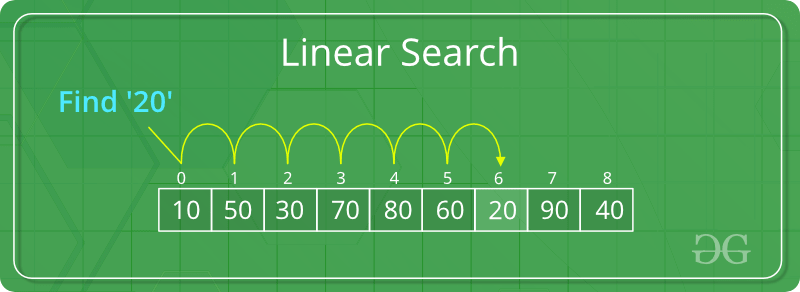


Figure: Example of sequential search

1. Interval Search: These algorithms are specifically designed for searching in sorted data-structures. These types 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, Ternary Search.

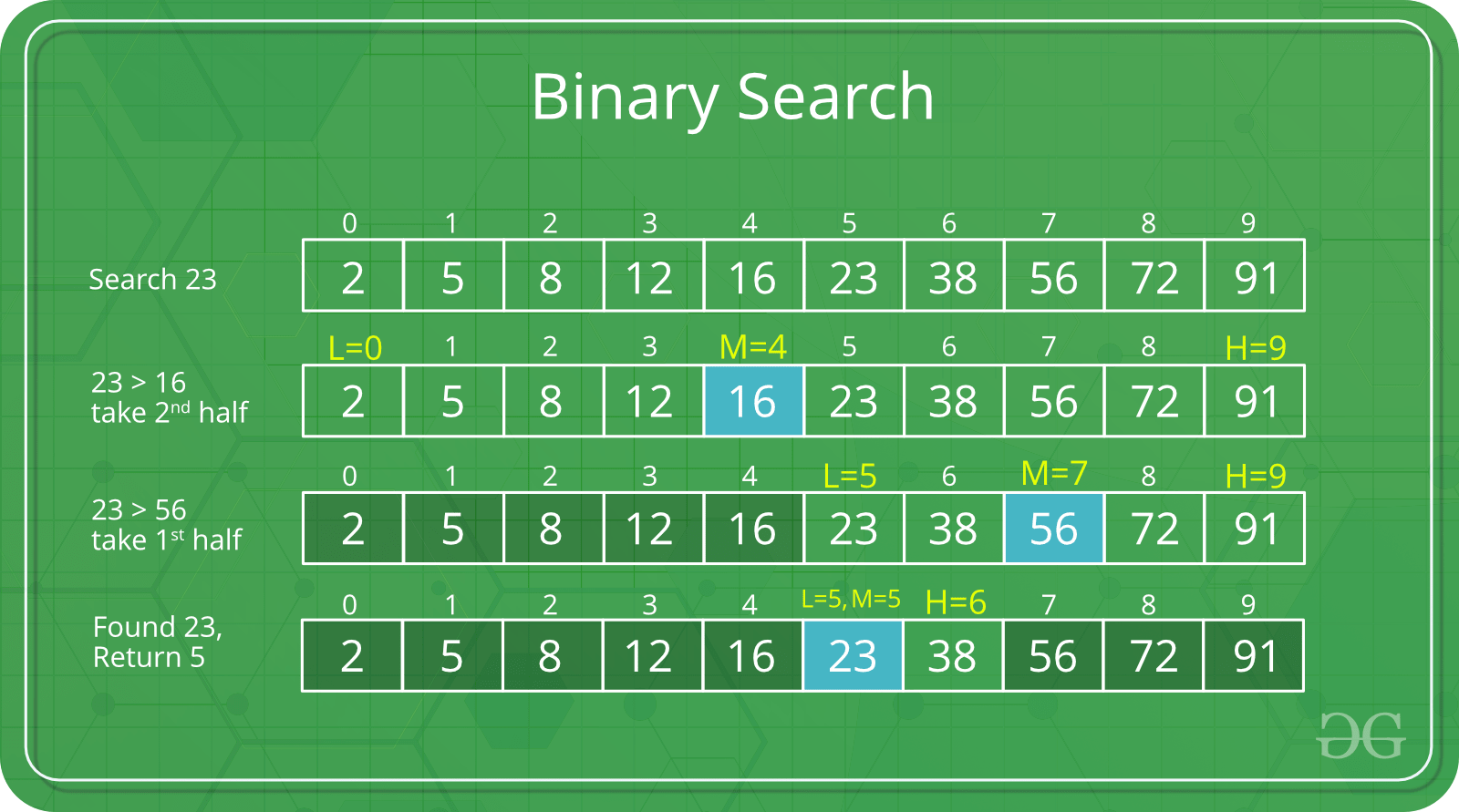


Figure: Example of interval search

Linear Search:

Linear search is used on a collection of items. It relies on the technique of traversing a list from start to end by exploring the properties of all the elements that are found on the way. For example, consider an array of integers of size N. You should find and print the position of all the elements with value x. Here, the linear search is based on the idea of matching each element from the beginning of the list to the end of the list with the integer x, and then printing the position of the element if the condition is true.

The pseudo code for this example is as follows :

| for(start to end of array)  {  if (current\_element equals to x)  {  print (current\_index);  }  } |
| --- |

Time and Space Complexity:

The time complexity of the linear search is O(N) because each element in an array is compared only once while the space complexity is O(N) as all the elements are initially stored in an array of size N.

Implementation:

|  |
| --- |

Binary Search:

Assignment

Ternary Search:

Assignment