

CSC508 Data Structures

Topic 11 : Searching Algorithms

Recap

- ▶ Selection Sort
- ▶ Insertion Sort
- ▶ Merge Sort
- ▶ Heap Sort

Topic Structure

- ▶ Sequential Search
- ▶ Binary Search

Learning Outcomes

- ▶ At the end of this lesson, students should be able to:
 - ▶ Describe Sequential and Binary searching algorithms
 - ▶ Implement Sequential and Binary search on lists

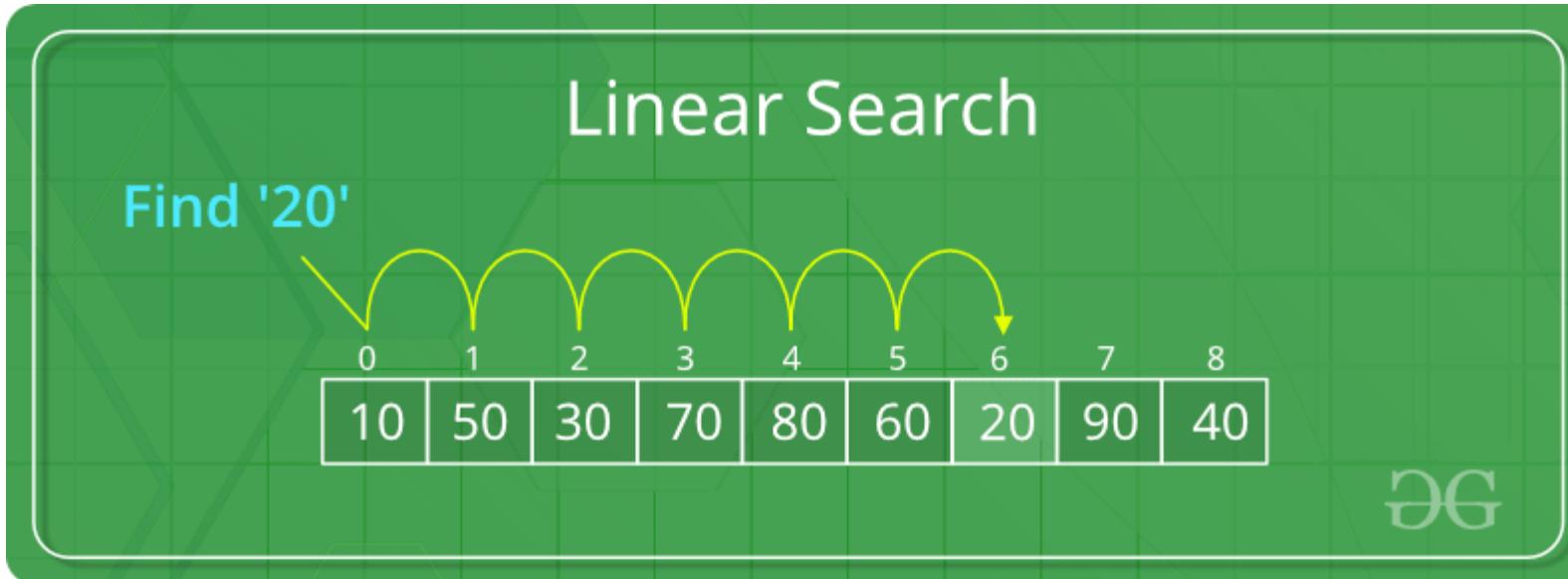
Searching

- ▶ Each element in a list should be identifiable by a one of its member (i.e. key).
 - ▶ For example, in list describing a student, the unique Student ID act as the key.
- ▶ Keys are used in such operations as searching, sorting, insertion, and deletion.
- ▶ Two general searching algorithms : Sequential and Binary search.

Sequential Search

- ▶ The ‘simplest’ searching algorithms implementing brute force approach.
- ▶ Check each element of the list one-by-one starting with the 1st element until the key is found or reach the end of list.
- ▶ Complexity : $O(n)$.
- ▶ Can be implemented in on both array and linked list.

Sequential Search Simulation



<https://www.geeksforgeeks.org/linear-search/>

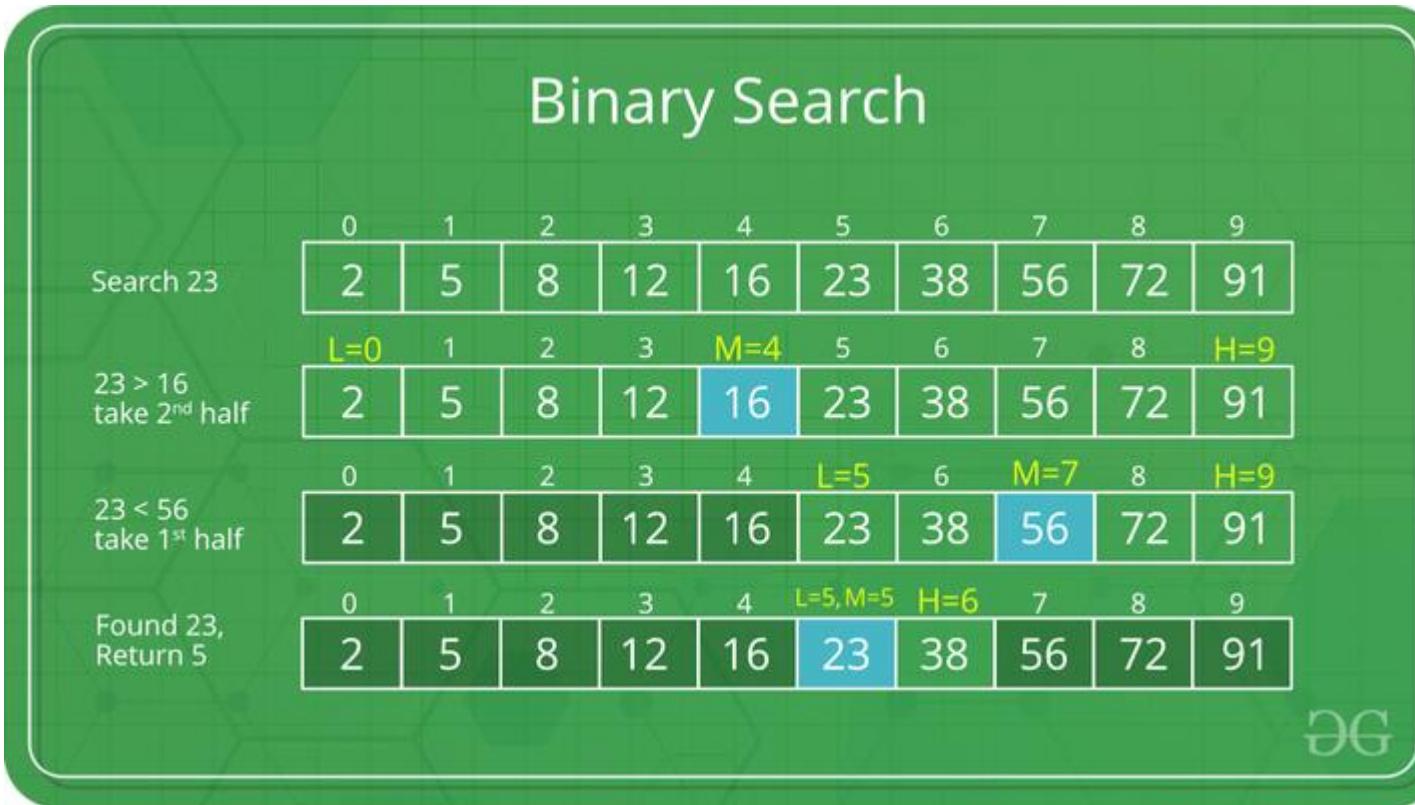
Sequential Search Implementation

```
public int seqSearch(Object searchItem) {  
    int loc;  
    boolean found = false;  
  
    for(loc = 0; loc < length; loc++)  
        if(list[loc].equals(searchItem)) {  
            found = true;  
            break; }  
  
    if(found)  
        return loc;  
    else  
        return -1;  
}
```

Binary Search

- ▶ Relatively fast searching algorithm, implementing divide and conquer approach.
- ▶ Starts by check the middle element.
 - ▶ Item is found if the search key is the middle element.
 - ▶ If the search key is less than the middle element, search the left section.
 - ▶ If the search key is greater than the middle element, search the right section.
- ▶ Elements in the list must be sorted.
- ▶ Complexity : $O(\log n)$

Binary Search Implementation



<https://www.geeksforgeeks.org/binary-search/>

Binary Search Implementation

```
public static void 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 ){  
            System.out.println("Element is found at index: " + mid);  
            break;  
        }else  
            last = mid - 1;  
  
        mid = (first + last)/2;  
    } //end while loop  
  
    if ( first > last )  
        System.out.println("Element is not found!");  
}
```

Summary

- ▶ A key is a unique identifier for element in list.
- ▶ Two basic searching algorithm :
 - ▶ Sequential search : Brute-force, $O(n)$
 - ▶ Binary search : Divide and Conquer, $O(\log n)$

Next Topic...

- ▶ Hash Map

References

- ▶ Carrano, F. & Savitch, W. 2005. *Data Structures and Abstractions with Java*, 2nd ed. Prentice-Hall.
- ▶ Malik D.S, & Nair P.S., Data Structures Using Java, Thomson Course Technology, 2003.
- ▶ Rada Mihalcea, CSCE 3110 Data Structures and Algorithm Analysis notes, U of North Texas.