SEQUENTIAL SEARCH

LINEAR SEARCH: is a sequential searching algorithm where we start from one end and check every element of the list until the desired element found. It is the simplest searching algorithm.

OR

It is a process of finding a given value position in a list of values.

* It is basic of simple search algorithm.

* In sequential search, we compare the target value with all the other elements given in the list.

eg:

start → 0 1 2 3 4

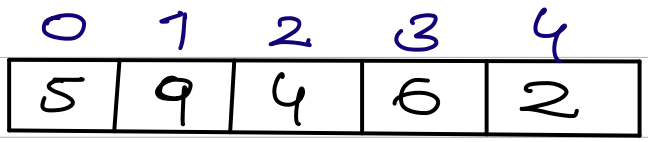
arr = [19, 12, 15, 23, 17]

Target = 23

In above example,

the target value is compared with all the elements in array in sequential/ linear way.

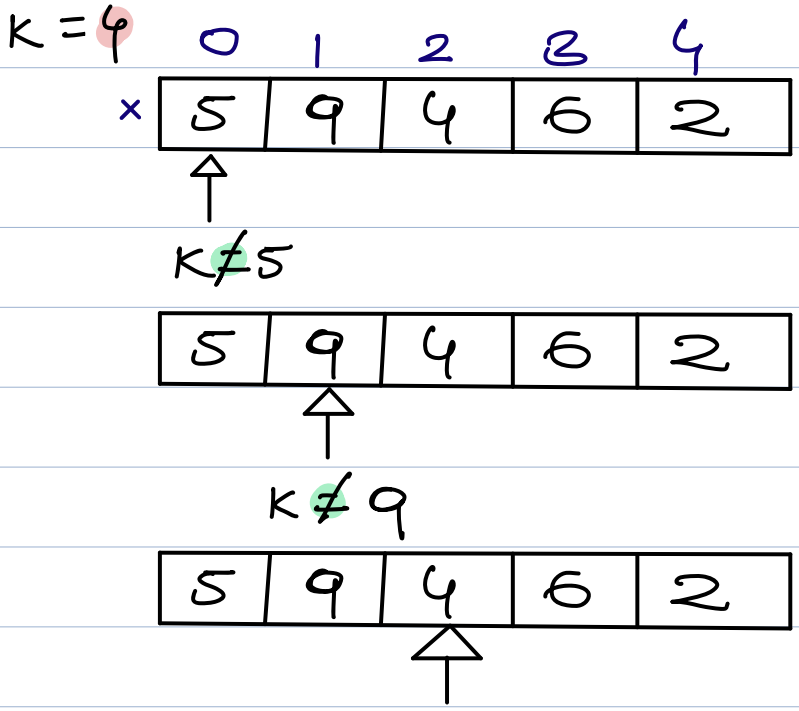
HOW LINEAR SEARCH WORKS:



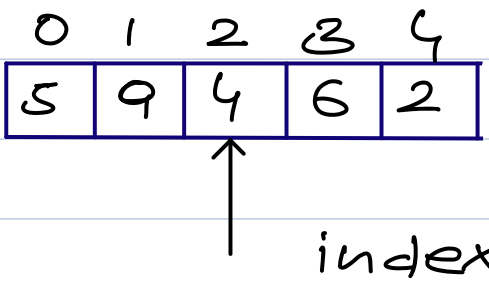
Array to be searched for

Search/Target = 4/ K=4

1. START
2. COMPARE elements with Array



2. If x == k, return the index



3. Else return not Found.

ALGORITHM :-

```
LinearSearch (int[] arr, target)
For each item in the array
        (FOR LOOP)
    int element = arr[index];
    if(element == target) {
        return index;
```

TIME COMPLEXITY

Best case : O(1) → constant

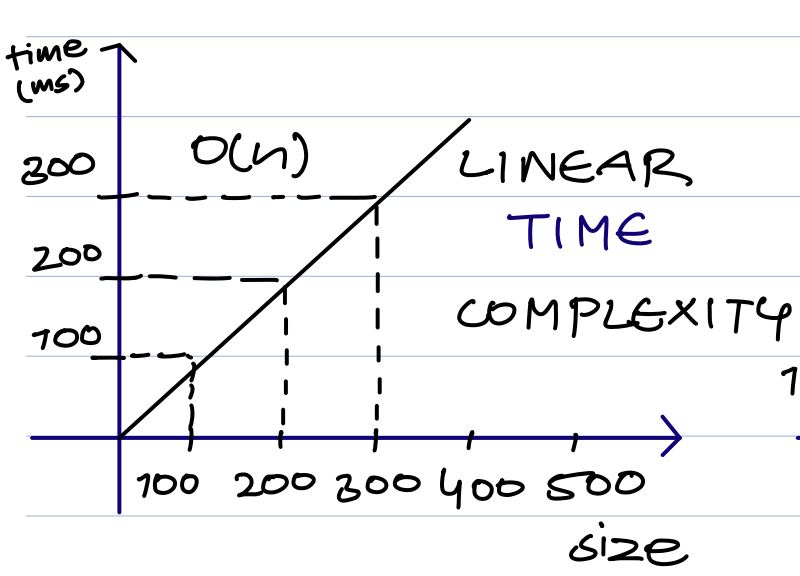
⇒ How many checks will the loop make in best case i.e, the element will be Found at 0TH index. only one comparison

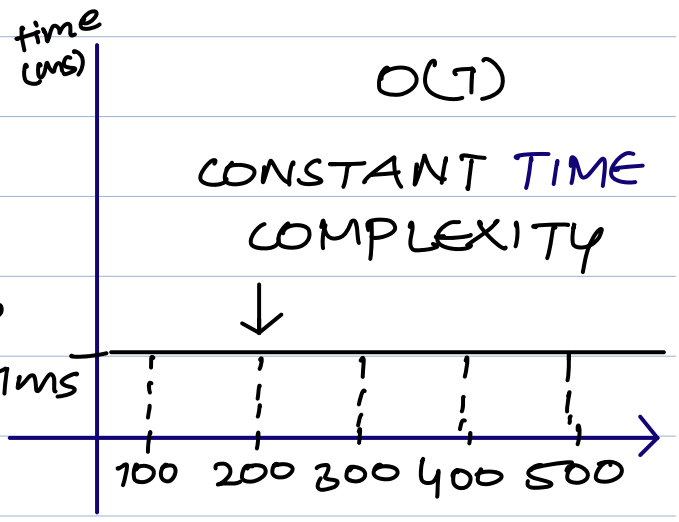
will be made for best case.

WORST CASE: O(n)

⇒ worst case, here it will go through every element and then it says element not found.

Size of Array	No of Comparisons	time(ms)
100	100	100ms
200	200	200ms
300	300	300ms
n	n	





WORST CASE

BEST CASE

PROGRAM

```java
public class LinearSearching {
    public static void main(String[] args) {
        int[] currency = {500,200,2000,100,50};
        int target = 2000;
        int ans = LinearSearch(currency, target);
        System.out.println("Target Element Index is " + ans + "th ");
        int ans2 = LinearSearch2(currency, target);
        System.out.println("Target Element " + ans2 + " is matched with Array");
     }
        //Function for returning index of the element
        static int LinearSearch(int[]arr, int target){
        if(arr.length == 0){
            return -1;
        }
        //For Loop
        for(int index = 0; index < arr.length; index++){
            int element  = arr[index];
            if (element == target){
                return index;
            }
        }
        return -1;
        }
       //Function for returning element
       static int LinearSearch2(int[] arr, int target){
        if(arr.length == 0){
            return -1;
        }
         for (int element : arr) {
            if (element == target) {
                return element;
            }
        }
        return -1;
    }
}
```

ASSIGNMENT

1. SEARCH IN RANGE
2. SEARCH IN STRING
3. FIND MIN
4. EVEN DIGITS

```java
//Serching element in the array but in range/
public class SearchInRange {
    public static void main(String[] args) {
        int[] arr = {20,30,10,12,15,18,19,21,23,25};
        int target = 18;
        int searchDone = find(arr,target,2,5);
        System.out.println(searchDone);
    }
    //Function
    //In this Program array range start from 2 and end into 5
    //If you target the element out of range then it will return -1
    static int find(int[]arr, int target, int start, int end){
        if(arr.length == 0){
            return -1;
        }
        for (int index = start; index <= end; index++){
            int element = arr[index];
            if(element == target){
                return index;
            }
        }
        return -1;
    }
    //SecondWay
    /*
      static int find(int[]arr, int target) {
        if (arr.length == 0) {
            return -1;
        }
        for (int index = 2; index <= 5; index++) {
            int element = arr[index];
            if (element == target) {
                return index;
            }
        }
        return -1;
    }
     */
}
```

```java
//Searching character in the String
public class SearchInString {
    public static void main(String[] args) {
        String name = "DhruvLenka";
        char target = 'v';
        System.out.println(search(name,target));

    }
    static boolean search(String str, char target){
        if(str.length() == 0){
            return false;
        }
        for(int i = 0; i <= str.length();i++){
            if(target == str.charAt(i)){
                return true;
            }
        }
        return false;
    }
}

```

```java
//Finding minimum num in the array
public class FindMin {
    public static void main(String[] args) {
        int[] arr = {17,12,18,5,-9,3,10,16,15,21};
        System.out.println(min(arr));
    }
    static int min(int[]arr){
        int MIN = arr[0];
        for (int i = 0; i <= MIN;i++) {
            if (arr[i] < MIN){
                MIN = arr[i];
            }
        }
        return MIN;
    }
}

```

```java
public class EvenDigits {
    public static void main(String[] args) {
        int[] nums = {12,345,2,6,7896,4567};
        System.out.println(findNumbers(nums));
    }
    static int findNumbers(int[] nums) {
        int count = 0;
        for(int num : nums) {
            if (even(num)) {
                count++;
            }
        }
        return count;
    }

    // function to check whether a number contains even digits or not
    static boolean even(int num) {
        int numberOfDigits = digits(num);
        /*
        if (numberOfDigits % 2 == 0) {
            return true;
        }
        return false;
         */
        return numberOfDigits % 2 == 0;
    }


    // count number of digits in a number
    static int digits(int num) {

        if (num < 0) {
            num = num * -1;
        }

        if (num == 0) {
            return 1;
        }

        int count = 0;
        while (num > 0) {
            count++;
            num = num / 10; // num /= 10
        }

        return count;
        /*
        if (num < 0) {
            num = num * -1;
        }
        return (int)(Math.log10(num)) + 1;
         */
    }
}
```