

## Linear Search

### Summary Table

Case	Time Complexity
Best Case	$O(1)$
Worst Case	$O(n)$

Search element in c++

```
#include <iostream>
#include <vector>
using namespace std;

// Function to perform linear search
int linearSearch(const vector<int>& arr, int target) {
    for (int i = 0; i < arr.size(); i++) {
        if (arr[i] == target) {
            return i; // Return index if found
        }
    }
    return -1; // Return -1 if not found
}

int main() {
    int n, target;
    //cout << "Enter the number of elements: ";
    cin >> n;

    vector<int> arr(n);
    //cout << "Enter the elements: ";
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    //cout << "Enter the target element to search: ";
    cin >> target;

    int result = linearSearch(arr, target);

    if (result != -1) {
        cout << "Element found at index " << result << endl;
    } else {
        cout << "Element not found in the array." << endl;
    }
    return 0;
}
```

Java

```
import java.util.Scanner;

public class LinearSearch {
    // Function to perform linear search
    public static int linearSearch(int[] arr, int target) {
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] == target) {
                return i; // Return index if found
            }
        }
        return -1; // Return -1 if not found
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        //System.out.print("Enter the number of elements: ");
        int n = sc.nextInt();

        int[] arr = new int[n];
        //System.out.println("Enter the elements: ");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        //System.out.print("Enter the target element to search: ");
        int target = sc.nextInt();

        int result = linearSearch(arr, target);

        if (result != -1) {
            System.out.println("Element found at index " + result);
        } else {
            System.out.println("Element not found in the array.");
        }

        sc.close();
    }
}
```

Linear search using recursion

```
#include <iostream>
#include <vector>
using namespace std;

// Recursive function for linear search
int linearSearchRecursive(const vector<int>& arr, int target, int index) {
    if (index == arr.size()) {
        return -1; // Base case: reached end of array, element not found
    }
}
```

```

    if (arr[index] == target) {
        return index; // Element found, return index
    }
    return linearSearchRecursive(arr, target, index + 1); // Recursive call
}

```

```

int main() {
    int n, target;
    //cout << "Enter the number of elements: ";
    cin >> n;

    vector<int> arr(n);
    //cout << "Enter the elements: ";
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    //cout << "Enter the target element to search: ";
    cin >> target;

    int result = linearSearchRecursive(arr, target, 0);

    if (result != -1) {
        cout << "Element found at index " << result << endl;
    } else {
        cout << "Element not found in the array." << endl;
    }

    return 0;
}

```

Java

```

import java.util.Scanner;

public class LinearSearchRecursive {
    // Recursive function for linear search
    public static int linearSearch(int[] arr, int target, int index) {
        if (index == arr.length) {
            return -1; // Base case: reached end of array, element not found
        }
        if (arr[index] == target) {
            return index; // Element found, return index
        }
        return linearSearch(arr, target, index + 1); // Recursive call
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        //System.out.print("Enter the number of elements: ");
        int n = sc.nextInt();
    }
}

```

```
int[] arr = new int[n];
//System.out.println("Enter the elements: ");
for (int i = 0; i < n; i++) {
    arr[i] = sc.nextInt();
}

//System.out.print("Enter the target element to search: ");
int target = sc.nextInt();

int result = linearSearch(arr, target, 0);

if (result != -1) {
    System.out.println("Element found at index " + result);
} else {
    System.out.println("Element not found in the array.");
}

sc.close();
}
```