

# *DAA\_Assign.2*

Name: Kaustubh Dhongade

Roll no.: 231070018

Batch: A

- Aim: Learn algorithm and time complexity of Binary and linear search

Code:

```
#include <bits/stdc++.h>
using namespace std;
#define ll long long

int linear_search(vector<int> arr, int key){
    for (int i = 0; i<arr.size(); i++){
        if (arr[i] == key){
            return i;
        }
    }
    return -1;
}

int binary_search(vector<int> arr, int key){

    int left = 0, right = arr.size()-1;
    sort(arr.begin(), arr.end());

    while (left <= right){
        int mid = left + (right - left)/2;
        if (arr[mid] == key){
            return mid;
        }

        if (arr[mid] < key) left = mid+1;
        else right = mid-1;
    }
}
```

```

        return -1;
    }

int main(){
    int n;
    cout << "Number of element in arr: ";
    cin >> n;
    vector<int> arr(n);
    cout << "Enter element: ";
    for (int i = 0; i<n; i++){
        cin >> arr[i];
    }

    int key;
    cout << "Enter the key to search: ";
    cin >> key;
    if (linear_search(arr, key) != -1) cout << "Number found at index (Linear Search): " << linear_search(arr, key) << endl;
    else cout << "Number not found" << endl;

    // if (binary_search(arr, key) != -1) cout << "Number found at index (Binary Search): " << binary_search(arr, key) << endl;
    // else cout << "Number not found" << endl;
}

```

## Conclusion:

*binary search is more efficient than linear search for sorted arrays, as it reduces the search space by half with each iteration, achieving a time complexity of  $O(\log n)$ .*

*Linear search, on the other hand, is simpler but less efficient, with a time complexity of  $O(n)$ , as it checks each element sequentially. While binary search is faster, it requires the data to be sorted, whereas linear search can be used on unsorted data.*

*The choice between the two depends on the nature of the data and the specific requirements of the application.*