



DEPARTMENT OF  
COMPUTER SCIENCE AND ENGINEERING

---

## Title: Implementation of Linear search and Binary search

---

DATA STRUCTURE LAB  
CSE 106



GREEN UNIVERSITY OF BANGLADESH

# 1 Objective(s)

- To gather knowledge on linear and binary search algorithms.
- To implement linear and binary search algorithms.
- To solve problems using linear and binary search algorithms.

## 2 Problem analysis

### 2.1 Linear Search

Linear Search is defined as a sequential search algorithm that starts at one end and goes through each element of a list until the desired element is found, otherwise the search continues till the end of the data set. In Linear Search Algorithm,

- Every element is considered as a potential match for the key and checked for the same.
- If any element is found equal to the key, the search is successful and the index of that element is returned.
- If no element is found equal to the key, the search yields “No match found”.

### 2.2 Binary Search

Binary Search is defined as a searching algorithm used in a sorted array by repeatedly dividing the search interval in half. The idea of binary search is to use the information that the array is sorted and reduce the time complexity to  $O(\log N)$ . To apply Binary Search algorithm:

- The data structure must be sorted.
- Access to any element of the data structure takes constant time.

## 3 Flowcharts

### 3.1 Linear Search

Here we discuss the linear search flowchart in which we start from the starting point check elements are present or it has zero element if it contains zero element then direct we can say that element not found else search element if found then print Element found at this position else increase the position by one and if all location have different then from the last position we can say element is not found. Below flowchart explain it in a clear way because vision clear all doubt easily.

### 3.2 Binary Search

Binary search is a vast improvement over the sequential search. For binary search to work, the item in the list must be in assorted order. The approach employed in the binary search is divide and conquer. If the list to be sorted for a specific item is not sorted, binary search fails. The flowchart of the iterative binary search is given below.

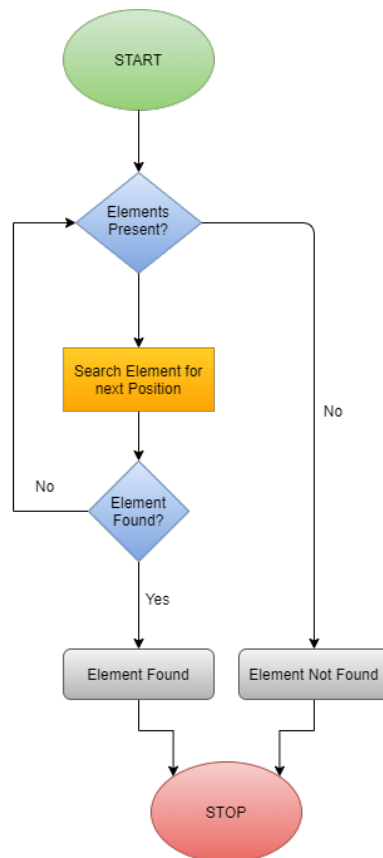


Figure 1: Flowchart of linear search.

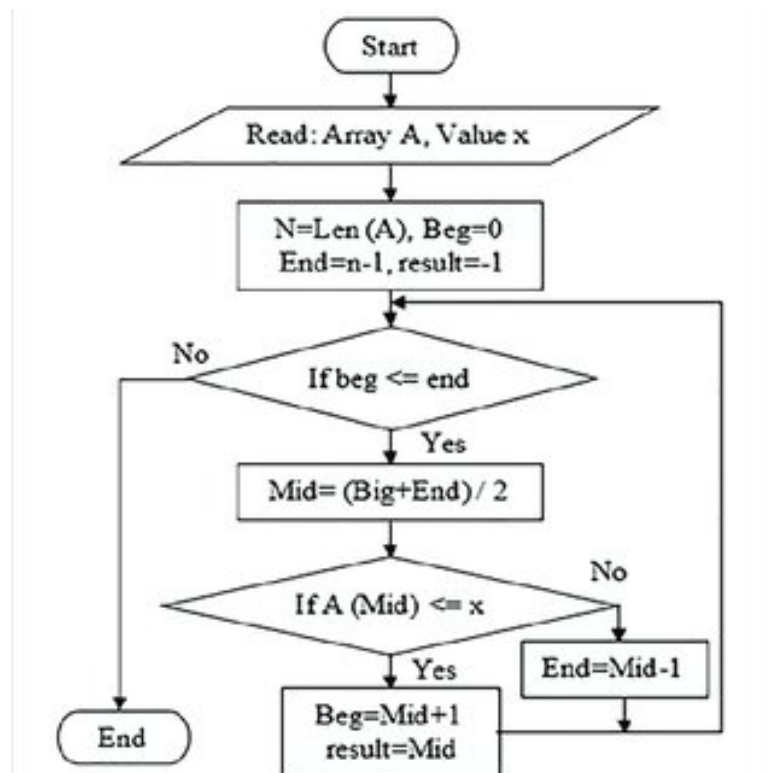


Figure 2: Flowchart of binary search.

## 4 Algorithms

### 4.1 Linear Search

The algorithm of linear search is given below.

---

**Algorithm 1:** Algorithm of linear search.

---

- 1 Linear Search ( Array A, Value x)
  - 2 Step 1: Set i to 1
  - 3 Step 2: if  $i > n$  then go to step 7
  - 4 Step 3: if  $A[i] = x$  then go to step 6
  - 5 Step 4: Set i to  $i + 1$
  - 6 Step 5: Go to Step 2
  - 7 Step 6: Print Element x Found at index i and go to step 8
  - 8 Step 7: Print element not found
  - 9 Step 8: Exit
- 

### 4.2 Binary Search

The algorithm of binary search is given below.

---

**Algorithm 2:** Algorithm of binary search.

---

- 1 Step 1 : Find the middle element of array. using ,  $middle = initial\_value + end\_value / 2$
  - 2 Step 2 : If  $middle = element$ , return 'element found' and index
  - 3 Step 3 : if  $middle > element$ , call the function with  $end\_value = middle - 1$
  - 4 Step 4 : if  $middle < element$ , call the function with  $start\_value = middle + 1$
  - 5 Step 5 : exit
- 

## 5 Implementation in C

### 5.1 Linear Search

```
1 #include <stdio.h>
2
3 int LINEAR_SEARCH(int inp_arr[], int size, int val)
4 {
5
6     for (int i = 0; i < size; i++)
7         if (inp_arr[i] == val)
8             return i;
9     return -1;
10 }
11
12
13 int main(void)
14 {
15     int arr[] = { 10, 20, 30, 40, 50, 100, 0 };
16     int key = 100;
17     int size = 10;
18     int res = LINEAR_SEARCH(arr, size, key);
19     if (res == -1)
20         printf("ELEMENT NOT FOUND!!");
21     else
22         printf("Item is present at index %d", res);
23
24     return 0;
25 }
```

### 5.1.1 Input/Output

Item is present at index 5

## 5.2 Binary Search

```
1  /* Binary Search in C */
2
3  #include <stdio.h>
4
5  int binarySearch(int array[], int x, int low, int high)
6  {
7      if (high >= low)
8      {
9          int mid = low + (high - low) / 2;
10
11         // If found at mid, then return it
12         if (array[mid] == x)
13             return mid;
14
15         // Search the left half
16         if (array[mid] > x)
17             return binarySearch(array, x, low, mid - 1);
18
19         // Search the right half
20         return binarySearch(array, x, mid + 1, high);
21     }
22
23     return -1;
24 }
25
26 int main(void)
27 {
28     int array[] = {3, 4, 5, 6, 7, 8, 9};
29     int n = sizeof(array) / sizeof(array[0]);
30     int x = 4;
31     int result = binarySearch(array, x, 0, n - 1);
32     if (result == -1)
33         printf("Not found");
34     else
35 }
```

### 5.2.1 Input/Output

Output of the given code is given below. Element is found at index 1

## 6 Discussion & Conclusion

Based on the focused objective(s) to understand about binary search, the additional lab exercise made me more confident towards the fulfilment of the objectives(s).

## 7 Lab Task (Please implement yourself and show the output to the instructor)

1. Implement linear search algorithms using recursion.

## 7.1 Problem analysis

Given an unsorted array and an element  $x$ , search  $x$  in given array. The idea is to compare  $x$  with first element in  $arr[]$ . If element is found at first position, return it. Else recur for remaining array and  $x$ .

## 7.2 Algorithm

---

**Algorithm 3:** Recursive Linear Search

---

- 1 Step 1: Program takes size of array.
  - 2 Step 2: Passing array, key and size to the recursive function `recursiveLinearSearch(int array[], int key, int size)`.
  - 3 Step 3: Recursive function calls itself until certain conditions fulfil.
  - 4 Step 4: Function returns 1 if record found in array else returns -1.
- 

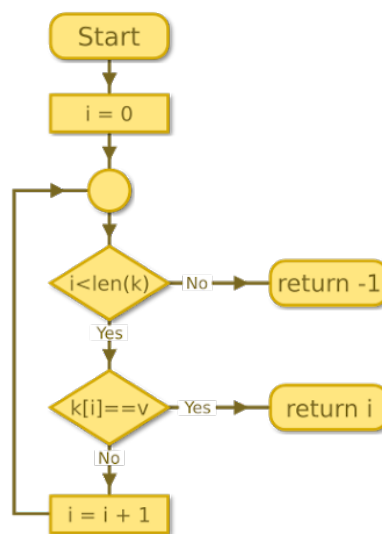


Figure 3: Flowchart of Linear Search

## 8 Lab Exercise (Submit as a report)

- Implement Linear Search for an array with character data using recursive method.
- Implement Binary Search for an array with character data using recursive method.

## 9 Policy

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.