



DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING

**Title: Implement Binary Search Algorithm using
Recursive Method**

DATA STRUCTURE LAB
CSE 106



GREEN UNIVERSITY OF BANGLADESH

1 Objective(s)

- To gather knowledge implement recursive methods in searching.
- To implement recursive linear and binary search algorithm.

2 Problem analysis

A recursive algorithm is an algorithm which calls itself with "smaller (or simpler)" input values, and which obtains the result for the current input by applying simple operations to the returned value for the smaller (or simpler) input. More generally if a problem can be solved utilizing solutions to smaller versions of the same problem, and the smaller versions reduce to easily solvable cases, then one can use a recursive algorithm to solve that problem. For example, the elements of a recursively defined set, or the value of a recursively defined function can be obtained by a recursive algorithm.

3 Flowchart

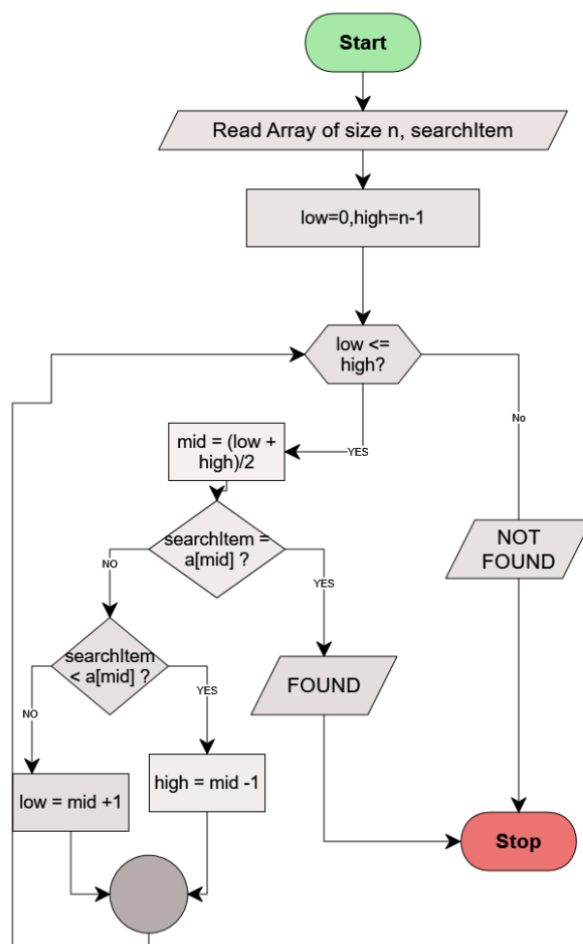


Figure 1: Flowchart of binary search.

4 Algorithm

Algorithm 1: Algorithm of recursive binary search.

Input: An Array, Value

- 1 Step 1: Compare x with the middle element.
 - 2 Step 2: If x matches with the middle element, we return the mid index.
 - 3 Step 3: Else If x is greater than the mid element, then x can only lie in the right half sub-array after the mid element. So we recur for the right half.
 - 4 Step 4: Else (x is smaller) recur for the left half.
-

5 Implementation in C

```
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 }
```

6 Input/Output

Output of the given code is given below.

Element is found at index 1

7 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).

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

1. Implement linear search algorithms using recursion.

8.1 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 .

8.2 Algorithm

Algorithm 2: 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.
-

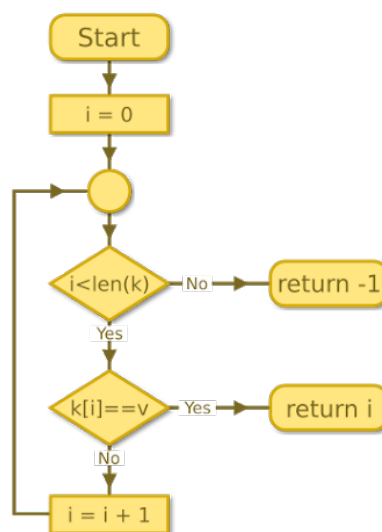


Figure 2: Flowchart of Linear Search

9 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.

10 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.