BÀI TẬP THỰC HÀNH CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

LAB MANUAL

Academic Year	:	**2022 - 2023**
Semester	:	**I**

Prepared by

MSc. KhangVQH

WEEK-1

SEARCHING TECHNIQUES

1.1 OBJECTIVE:

a. Write a C program to for implementing linear search technique.
b. Write a C program to for implementing binary search technique.

1.2 PROGRAM LOGIC:

Linear Search Algorithm

```
Algorithm linsrch (a[], x)
 { // a[1:n] is an array of n elements
      index := 0; flag := 0;
           while (index < n) do
           {
                    if (x = a[index]) then
                    {
                              flag := 1;
                              break;
                    }
                   index++;
           }
           if(flag =1)
                    write("Data found ");
           else
                    write("data not found");
}
```

Example: Given a list of n elements and search a given element x in the list using linear search.

a. Start from the leftmost element of list a[] and one by one compare x with each element of list a[].
b. If x matches with an element, return the index.
c. If x doesn't match with any of elements, return -1.

Consider a list with 10 elements and search for 9.

a = [56, 3, 249, 518, 7, 26, 94, 651, 23, 9]

Index →	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Iteration 1	56	3	249	518	7	26	94	651	23	9
Iteration 2	56	3	249	518	7	26	94	651	23	9
Iteration 3	56	3	249	518	7	26	94	651	23	9
Iteration 4	56	3	249	518	7	26	94	651	23	9
Iteration 5	56	3	249	518	7	26	94	651	23	9
Iteration 6	56	3	249	518	7	26	94	651	23	9
Iteration 7	56	3	249	518	7	26	94	651	23	9
Iteration 8	56	3	249	518	7	26	94	651	23	9
Iteration 9	56	3	249	518	7	26	94	651	23	9
Iteration 10	56	3	249	518	7	26	94	651	23	9

Binary Search Algorithm

```
Algorithm binsrch (a[], n, x)
 { // a[1:n] is an array of n elements
       low = 1;
       high = n;
       while (low < high) do
       {
             mid = (low + high)/2 ;
             if (x < a[mid]) then
                          high = mid – 1;
             else if (x > a[mid]) then
                          low = mid + 1;
              else
                          return mid;
        }
        return 0;
}
```

Example: Given a sorted list of a[] of n elements, search a given element x in list.

a. Search a sorted list by repeatedly dividing the search interval in half. Begin with an interval covering the whole list.
b. If the search key is less than the item in the middle item, then narrow the interval to the lower half. Otherwise narrow it to the upper half.
c. Repeat the procedure until the value is found or the interval is empty.

Consider a sorted list a[] with 9 elements and the search key is 31.

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
11	23	31	33	65	68	71	89	100

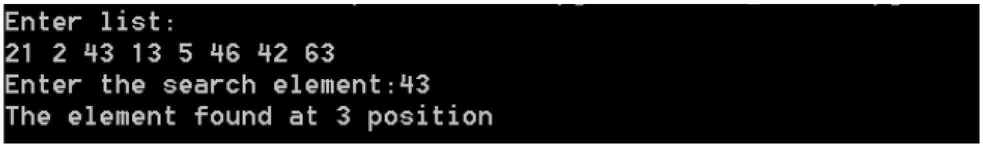
Let the search key = 31.
First low = 0, high = 8, mid = (low + high) = 4
a[mid] = 65 is the centre element, but 65 > 31.
So now high = mid - 1= 4 - 1 = 3, low = 0, mid = (0 + 3) / 2=1
a[mid] = a[1] = 23, but 23 < 31.

Again low = mid +1 = 1 +1 =2, high = 3, mid = (2 + 3) /2 = 2
a[mid] = a[2] = 31 which is the search key, so the search is successful.

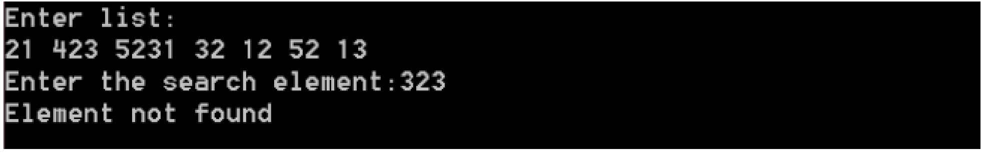
1.3 IMPLEMENTATION:

- Implementation of Linear Search

OUTPUT:



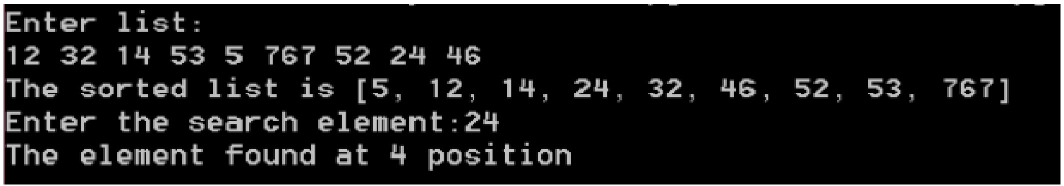
```
Enter list:
21 2 43 13 5 46 42 63
Enter the search element:43
The element found at 3 position
```



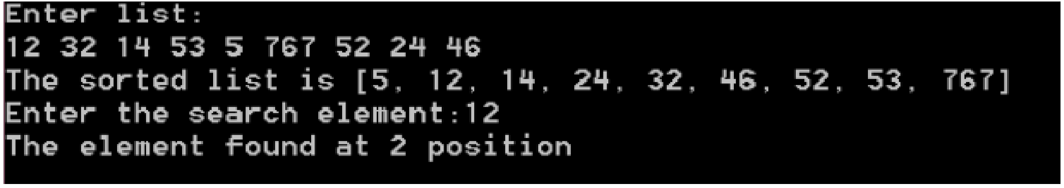
```
Enter list:
21 423 5231 32 12 52 13
Enter the search element:323
Element not found
```

- Implementation of Binary Search

OUTPUT:



```
Enter list:
12 32 14 53 5 767 52 24 46
The sorted list is [5, 12, 14, 24, 32, 46, 52, 53, 767]
Enter the search element:24
The element found at 4 position
```



```
Enter list:
12 32 14 53 5 767 52 24 46
The sorted list is [5, 12, 14, 24, 32, 46, 52, 53, 767]
Enter the search element:12
The element found at 2 position
```

1.4 LAB ASSIGNMENT:

1. A person has registered for voter id , he received a voter number and he need to check whether it exist in the voter or not. Use a binary searching in a recursive way to find whether the voter number exist in the list or not.
2. Use linear search technique to search for a key value in a given list of characters and print the message found or not.
3. A structure is defined as follows.

```
struct Record {
        int RollNo;
        char name[20];
        struct Date { int dd; int mm; int yy; } dob;
}
```

suppose, 100 records of type Record are stored in an array say, struct Record students[100];
We have to find the student(s), whose date of birth is given in the form **dd/mm/yy**. How you can search the array students?

1.5 POST LAB QUESTIONS:

Find the time complexity of linear search?
Find the time complexity of binary search?