06/09/2020

**Experiment No. 4**

**LINEAR SEARCH**

**AIM:**

To find an element in a given array using the linear search algorithm

**DATA STRUCTURES USED:**

Arrays

**ALGORITHM:**

Step 1: Set i to 1

Step 2: if i > n then go to step 7

Step 3: if A[i] = x then go to step 6

Step 4: Set i to i + 1

Step 5: Go to Step 2

Step 6: Print Element x Found at index i and go to step 8

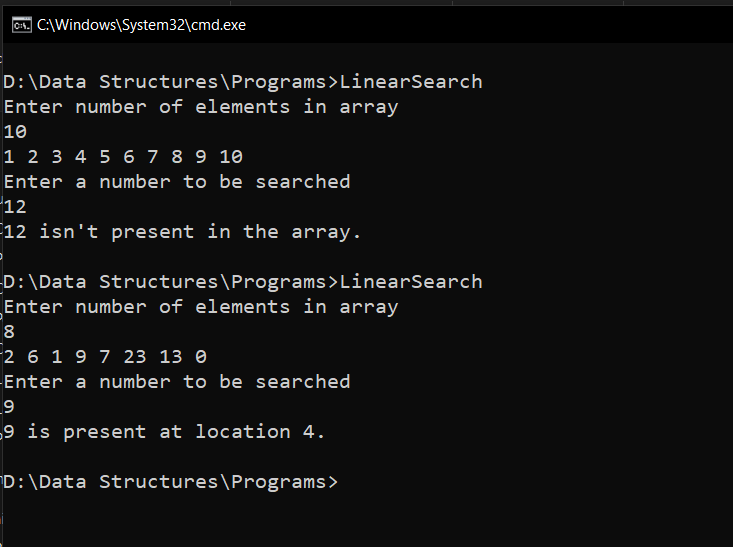
Step 7: Print element not found

Step 8: Exit

**PROGRAM:**

#include <stdio.h>  
  
int main()  
{  
 int arr[100], search, i, n;  
  
 printf("Enter number of elements in array\n");  
 scanf("%d", &n);  
  
 for (i = 0; i < n; i++){  
 scanf("%d", &arr[i]);  
 }  
 printf("Enter a number to be searched \n");  
 scanf("%d", &search);  
  
 for (i = 0; i < n; i++)  
 {  
 if (arr[i] == search)  
 {  
 printf("%d is present at location %d.\n", search, i+1);  
 break;  
 }  
 }  
 if (i == n)  
 printf("%d isn't present in the array.\n", search);  
  
 return 0;  
}

**OUTPUT:**



**RESULT:**

The Program was successfully compiled and the required output was

obtained

The Best case time complexity is O(1)

The Worst case time complexity is O(n)

Experiment No. 5

BINARY SEARCH

06/09/2020

AIM :

To find an element in a give array using the binary search algorithm

DATA STRUCTURES USED:

Arrays

ALGORITHM:

Step 1: Select the element in the middle of the array.

Step 2: Compare the selected element to the searched element, if it is

equal to the searched element, terminate.

Step 3: If the searched element is larger than the selected element, repeat

the search operation in the major part of the selected element.

Step 4: If the searched element is smaller than the selected element, repeat

the search in the smaller part of the selected element.

Step 5 : Repeat the steps until the smallest index in the search space is less

than or equal to the largest index

PROGRAM:

#include <stdio.h>  
  
int main()  
{  
 int i, first, last, middle, n, search, arr[100];  
  
 printf("Enter number of elements\n");  
 scanf("%d", &n);  
 for (i = 0; i < n; i++){  
 scanf("%d", &arr[i]);  
 }  
 printf("Enter element to searched\n");  
 scanf("%d", &search);  
  
 first = 0;  
 last = n-1;  
 middle = (first+last)/2;  
  
 while (first <= last) {  
 if (arr[middle] < search){  
 first = middle + 1;  
 }  
 else if (arr[middle] == search) {  
 printf("%d found at location %d\n", search, middle+1);  
 break;  
 }  
 else{  
 last = middle - 1;  
 }  
 middle = (first + last)/2;  
 }  
 if (first > last){  
 printf("Not found! %d isn't present \n", search);  
 }  
 return 0;  
}

RESULT:

The Program was successfully compiled and the required output was

obtained

The Best case time complexity is O(1)

The Worst case time complexity is O(log(n))

OUTPUT:

