

**Symbiosis Institute of Technology**

**Faculty of Engineering**

**CSE- Academic Year 2023-24**

**Data Structures – Lab Batch 2022-26**

|  |  |
| --- | --- |
| **Lab Assignment No:- 1,2,3** | |
|  | |
| **Name of Student** |  |
| **PRN No.** |  |
| **Batch** |  |
| **Class** |  |
| **Academic Year & Semester** |  |
| **Date of Submission** |  |
|  | |
| **Title of Assignment:** | A. Implement following searching algorithm: Linear search with multiple occurrences  B. Implement following searching algorithms in menu:  1. Binary search with iteration  2. Binary search with recursion |
| **Theory:** | 1. Prepare table for following searching and sorting algorithms for their best case, average case and worst case time complexities.   Linear search, binary search, bubble sort, Insertion sort, selection sort, merge sort, quick sort.   1. Discuss on Best case and Worst case time complexities of   Linear search, binary search, bubble sort, Insertion sort, selection sort, merge sort, quick sort. |
| **Source Code/Algorithm/Flow Chart:** | Linear Search Code :  #include<stdio.h>  int main()  {  int flag=0;  int n;  printf("Enter the no. of elements");  scanf("%d",&n);  int a[n];  for(int i=0;i<n;i++)  {  printf("Enter array element:");  scanf("%d",&a[i]);  }  int data;  printf("Enter the element to be searched:");  scanf("%d",&data);  for(int i=0;i<n;i++)  {  if(a[i]==data)  {  flag=1;  printf("Element found at %d index",i);  }  }  if(flag==0)  {  printf("Element not found");  }  return 0;  }  Code with Binary search :  #include<stdio.h>  int binarysearch(int a[] , int n ,int key){  int l=0;  int r = n-1;  while(l<n){  int mid = (l+r)/2;  if(a[mid]==key){  return mid;  }  else if(key<a[mid]){  r = mid - 1;  }  else{  l = mid + 1;  }  }  return -1 ;  }  int main(){  int a [] = {5,7,8,9,75,47,55,87};  int key = 55;  int loc,n;  n = sizeof(a)/sizeof(a[0]);  loc = binarysearch(a,n,key);  printf("Key number of %d is in place %d index",key,loc);  }  Code with Recursion :  // Online C compiler to run C program online  #include <stdio.h>  int binarysearch(int a[],int n,int key){  int l = 0;  int r = n-1;  while(l<n){  int mid = (l+r)/2;  if(a[mid]==key){  return mid;  }  else if(a[mid]<key){  binarysearch(a,0,mid - 1);  }  else{  binarysearch(a,mid + 1,r);  }  return -1;  }  }  int main() {  int a[] = {7,8,9,74,6,54,23};  int n = sizeof(a)/sizeof(a[0]);  int key = 74;  int loc = binarysearch(a,n,key);  printf("Key number of %d is in place %d index",key,loc);  return 0;  } |
| **Output Screenshots (if applicable)** | Liner Search Code :  Binary Search Code: |
| **Conclusion** | Thus we have studied different sorting algorithms and their time complexities. |