INDIAN INSTITUTE OF INFORMATION TECHNOLOGY BHOPAL

DAA LAB

Name: Tanav Singh Bajaj
Scholar No. : 20U03037
Semester: Third
Branch: IT

Assignment 2

Ans 1

Linear Search Linear search is the simplest search algorithm and often called sequential search. In this type of searching, we simply traverse the list completely and match each element of the list with the item whose location is to be found. If the match is found then the location of the item is returned otherwise the algorithm returns NULL.
Binary Search Binary search is the search technique which works efficiently on sorted lists. Hence, in order to search an element into some list by using binary search technique, we must ensure that the list is sorted. Binary search follows a divide and conquer approach in which the list is divided into two halves and the item is compared with the middle element of the list. If the match is found then, the location of the middle element is returned; otherwise, we search into either of the halves depending upon the result produced through the match.

Ans 2

Linear Search

A simple approach is to do a linear search, i.e

1. Start from the leftmost element of arr[] and one by one compare x with each element of arr[ ]
2. If x matches with an element, return the index.
3. If x doesn't match with any of elements, return -1
4. **Time Complexity:** O(n)
5. **Space Complexity:** O(1)

Binary Search

Search a sorted array by repeatedly dividing the search interval in half. Begin with an interval covering the whole array. If the value of the search key is less than the item in the middle of the interval, narrow the interval to the lower half. Otherwise, narrow it to the upper half. Repeatedly check until the value is found or the interval is empty

Calculating Time complexity:

Let say the iteration in Binary Search terminates after k iterations. Let's take an example where it terminates after 3 iterations, so here k = 3

At each iteration, the array is divided by half.

let's say the length of array at any iteration is n

At Iteration 1,

Length of array = n

At Iteration 2,

Length of array = $n/2$

At Iteration 3,

Length of array = $n/4$

Therefore, **after Iteration k,**

Length of array = $n/2^k$

Also, we know that after

After k divisions, the length of array becomes 1

Now,
Length of array = $n/2^k = 1$

$\Rightarrow n = 2^k$

Applying log function on both sides:
$\Rightarrow \log_2(n) = \log_2(2^k)$
$\Rightarrow \log_2(n) = k \log_2(2)$

As $(\log_a(a) = 1)$

Therefore,

$\Rightarrow k = \log_2(n)$

Hence Time complexity of Binary Search is log2(n)

Ans 3

Linear Search

```
#include <bits/stdc++.h>
using namespace std;

int linearSearch(int arr[], int n, int x);

int main()
{
    int t;
    cout << "enter length of array \n";
    cin >> t;
    int arr[t];
    for (int i = 0; i < t; i++)
    {
        cout << "enter value at position  " << i << endl;
        cin >> arr[i];
    }

    int x;
    cout << "Enter a number" << endl;
    cin >> x;
    int n = sizeof(arr) / sizeof(arr[0]);
    int result = linearSearch(arr, n, x);
    if (result == -1)
    {
        cout << "Element is not present in array \n";
    }
    else
    {
        cout << "Element is present at " << result << " position \n";
    }
    cout << "Coder name = Tanav Singh Bajaj \n";
    cout << "Coder Scholar Number= 20U03037 \n";
    return 0;
}
```

```
int linearSearch(int arr[], int n, int x)
{
    for (int i = 0; i < n; i++)
    {
        if (arr[i] == x)
        {
            return i;
        }
    }
    return -1;
}
```

```
enter length of array
6
enter value at position  0
1
enter value at position  1
2
enter value at position  2
4
enter value at position  3
5
enter value at position  4
7
enter value at position  5
2
Enter a number
7
Element is present at 4 position
Coder name = Tanav Singh Bajaj
Coder Scholar Number= 20U03037
```

```
enter length of array
5
enter value at position  0
2
enter value at position  1
3
enter value at position  2
7
enter value at position  3
2
enter value at position  4
7
Enter a number
9
Element is not present in array
Coder name = Tanav Singh Bajaj
Coder Scholar Number= 20U03037
```

Ans 4

Binary Search with Recursion

```
#include <iostream>
using namespace std;
int binarySearch(int arr[], int l, int r, int x);

int main()
{
    int t;
    cout << "enter length of array \n";
    cin >> t;
    int arr[t];
    cout << "enter values in ascending order for binary search to
work properly \n";
    for (int i = 0; i < t; i++)
    {
        cout << "enter value at position  " << i << endl;
        cin >> arr[i];
    }
    int x;
```

```
    cout << "Enter a number" << endl;
    cin >> x;
    int n = sizeof(arr) / sizeof(arr[0]);
    int result = binarySearch(arr, 0, n - 1, x);
    if (result == -1)
    {
        cout << "Element is not present in array" << endl;
    }
    else
    {
        cout << "Element is present at position  " << result << endl;
    }
    cout << "Coder name = Tanav Singh Bajaj \n";
    cout << "Coder Scholar Number= 20U03037 \n";
    return 0;
}
int binarySearch(int arr[], int l, int r, int x)
{
    if (r >= l)
    {
        int mid = l + (r - l) / 2;
        if (arr[mid] == x)
        {
            return mid;
        }
        else if (arr[mid] > x)
        {
            return binarySearch(arr, l, mid - 1, x);
        }
        else
        {
            return binarySearch(arr, mid + 1, r, x);
        }
    }
    return -1;
}
```

```
enter length of array
4
enter values in ascending order for binary search to work properly
enter value at position  0
2
enter value at position  1
3
enter value at position  2
4
enter value at position  3
6
Enter a number
3
Element is present at position  1
Coder name = Tanav Singh Bajaj
Coder Scholar Number= 20U03037
```

```
enter length of array
3
enter values in ascending order for binary search to work properly
enter value at position  0
4
enter value at position  1
5
enter value at position  2
7
Enter a number
1
Element is not present in array
Coder name = Tanav Singh Bajaj
Coder Scholar Number= 20U03037
```

Ans 5

```cpp
#include <iostream>
using namespace std;
int binarySearch(int arr[], int l, int r, int x);

int main()
{
    int t;
    cout << "enter length of array \n";
    cin >> t;
    int arr[t];
    for (int i = 0; i < t; i++)
    {
        cout << "enter value at position  " << i << endl;
        cin >> arr[i];
    }
    int x;
    cout << "Enter a number" << endl;
    cin >> x;
    int n = sizeof(arr) / sizeof(arr[0]);
    int result = binarySearch(arr, 0, n - 1, x);
    if (result == -1)
    {
        cout << "Element is not present in array" << endl;
    }
    else
    {
        cout << "Element is present at position " << result << endl;
    }
    cout << "Coder name = Tanav Singh Bajaj \n";
    cout << "Coder Scholar Number= 20U03037 \n";
    return 0;
}

int binarySearch(int arr[], int l, int r, int x)
{
    while (l <= r)
    {
        int mid = l + (r - l) / 2;

        if (arr[mid] == x)
```

```
        {
            return mid;
        }
        if (arr[mid] < x)
        {
            l = mid + 1;
        }

        else
        {
            r = mid - 1;
        }
    }
    return -1;
}
```

```
enter length of array
6
enter value at position  0
2
enter value at position  1
3
enter value at position  2
4
enter value at position  3
5
enter value at position  4
6
enter value at position  5
7
Enter a number
4
Element is present at position 2
Coder name = Tanav Singh Bajaj
Coder Scholar Number= 20U03037
```