**Practical-5**

1. Implement a function of binary search and count the steps executed by function on various inputs for best case and worst case. Also write complexity in each case and draw a comparative chart.

**Code:**

#include <bits/stdc++.h>

using namespace std;

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;

if (arr[mid] > x)

return binarySearch(arr, l, mid - 1, x);

return binarySearch(arr, mid + 1, r, x);

}

return -1;

}

int main(void)

{

int arr[] = { 2, 3, 4, 10, 40 };

int x = 10;

int n = sizeof(arr) / sizeof(arr[0]);

int result = binarySearch(arr, 0, n - 1, x);

(result == -1)

? cout << "Element is not present in array"

: cout << "Element is present at index " << result;

return 0;

}

Output:

