Practical - 4

**Implement a function of sequential 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 <stdio.h>

void Sequential\_search(int n,int a[],int search)

{

int i, count=0; for(i=0;i<n;i++)

{

count++; if(a[i]==search)

{

printf("Searched element found at index : %d",i); break;

}

}

if(i==n)

{

printf("Searched element not found!!");

}

printf("\nCounter value : %d",count);

}

int main(void) {

int arr[]={12,2,44,76,5,23,31,11,7,10};

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

scanf("%d",&s); Sequential\_search(n,arr,s); return 0;

}

# Observation :-

* **BEST CASE :-**

|  |  |
| --- | --- |
| **N-values** | **Time complexity** |
| 10 | 0.006324 |
| 20 | 0.00599 |
| 30 | 0.007201 |
| 40 | 0.006439 |
| 50 | 0.006407 |

* **Conclusion:** In Best case the Time complexity : O(1).

# WORST CASE :-

|  |  |
| --- | --- |
| **N-values** | **Time complexity** |
| 10 | 0.005998 |
| 20 | 0.006891 |
| 30 | 0.006453 |
| 40 | 0.006345 |
| 50 | 0.007123 |

* **Conclusion:** In Worst case the Time complexity : O(n).

# Best case v/s worst case :-

**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 <stdio.h>

int binarySearch(int array[], int x, int low, int high) { while (low <= high) {

int mid = low + (high - low) / 2;

if (array[mid] == x) return mid;

if (array[mid] < x) low = mid + 1;

else

high = mid - 1;

}

return -1;

}

int main(void) { int n =10

int array[] = {};

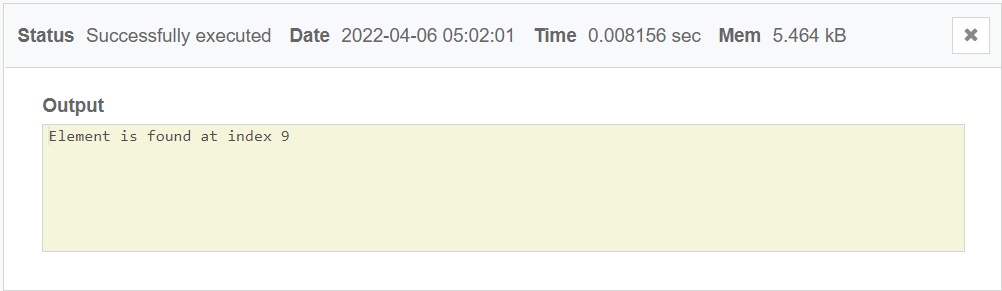
int x = 4;

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

if (result == -1) printf("Not found"); else

printf("Element is found at index %d", result); return 0;

}



Best Case :-

|  |  |
| --- | --- |
| No.of Elements | Time |
| 10 | 0.006000 |
| 20 | 0.006432 |
| 30 | 0.007043 |
| 40 | 0.007459 |
| 50 | 0.007986 |

Time Complexity: O(1)

Worst Case :-

|  |  |
| --- | --- |
| No of Elements | Time |
| 10 | 0.008111 |
| 20 | 0.008645 |
| 30 | 0.009018 |
| 40 | 0.010556 |
| 50 | 0.011232 |

Time Complexity: O(logn)

Best Case Vs Worst Case :-

Conclusion :-

For Binary search best case will be when key element(element to be searched) is first element of the array and time complexity will be O(1)

And worst case will be key element is last element or not present in array in that case time complexity will be O(logn)