

DSA_lab assignment

Mayank Gupta
20BCE1538

AIM Implementation of Searching Algorithms

A) Linear Search:

```
#include <stdio.h>

//linear search function
int search(int arr[], int n, int x) {

    int i;

    for (i = 0; i < n; i++)

        if (arr[i] == x) return i;

    return -1;

}

int main(void) {

    int arr[] = {15, 45, 8, 65, 52, 45, 65, 69, 12, 10};

    int x; scanf("%d", &x); int n = sizeof(arr) / sizeof(arr[0]);

    // Function call

    int result = search(arr, n, x);

    if (result == -1) printf("Element is not present in array");

    else printf("Element is present at position %d", result + 1);

    return 0;

}
```

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

A screenshot of a console window with a dark background. The window has a title bar with standard icons (minimize, maximize, close) and a blue vertical bar on the left side. The text inside the console is as follows:

```
45  
Element is present at position 2  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

B

B) Binary Search

```
#include <stdio.h>
```

```
//linear search function
int binarySearch(int a[], int s, int e, int f) {
    int m;

    if (s > e) // Not found
        return -1;

    m = (s + e)/2;

    if (a[m] == f) // element found
        return m;
    else if (f > a[m])
        return binarySearch(a, m+1, e, f);
    else
        return binarySearch(a, s, m-1, f);
}

int main()
{
    int c, first, last, n, search, array[100], index;

    printf("Enter number of elements\n");
    scanf("%d", &n);

    printf("Enter %d integers\n", n);

    for (c = 0; c < n; c++)
        scanf("%d", &array[c]);

    printf("Enter value to find\n");
    scanf("%d", &search);

    first = 0;
    last = n - 1;

    index = binarySearch(array, first, last, search);

    if (index == -1)
        printf("Not found! %d isn't present in the list.\n", search);
    else
        printf("%d is present at location %d.\n", search, index + 1);

    return 0;
}
```

```
14 return binarySearch(a, m+1, e, f);  
Enter number of elements  
5  
Enter 5 integers  
1  
5  
3  
5  
3  
Enter value to find  
3  
3 is present at location 3.  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

ALGORITHM

1. Step 1: set **beg** = **lower_bound**, **end** = **upper_bound**, **pos** = - 1
2. Step 2: repeat steps 3 and 4 while **beg** <= **end**
3. Step 3: set **mid** = (**beg** + **end**)/2
4. Step 4: if **a[mid]** = **val**
5. set **pos** = **mid**
6. print **pos**
7. go to step 6
8. else if **a[mid]** > **val**
9. set **end** = **mid** - 1
10. else
11. set **beg** = **mid** + 1
12. [end of if]
13. [end of loop]
14. Step 5: if **pos** = -1
15. print "value is not present in the array"
16. [end of if]
17. Step 6: exit

