Summary DSA

Sessions No 06(14-12-2022)

- Traverse into the array is known as **Traversal**. For traversing in an array we need a loop and for that we should know from where we start and till where we will go.

- As of now we know from where we need start ie.0 but data size in array is vary therefore we need to find till where we need to go ie. size of array.

- For example,

  int data[] = {5, 2, 7, 3, 9, 1};

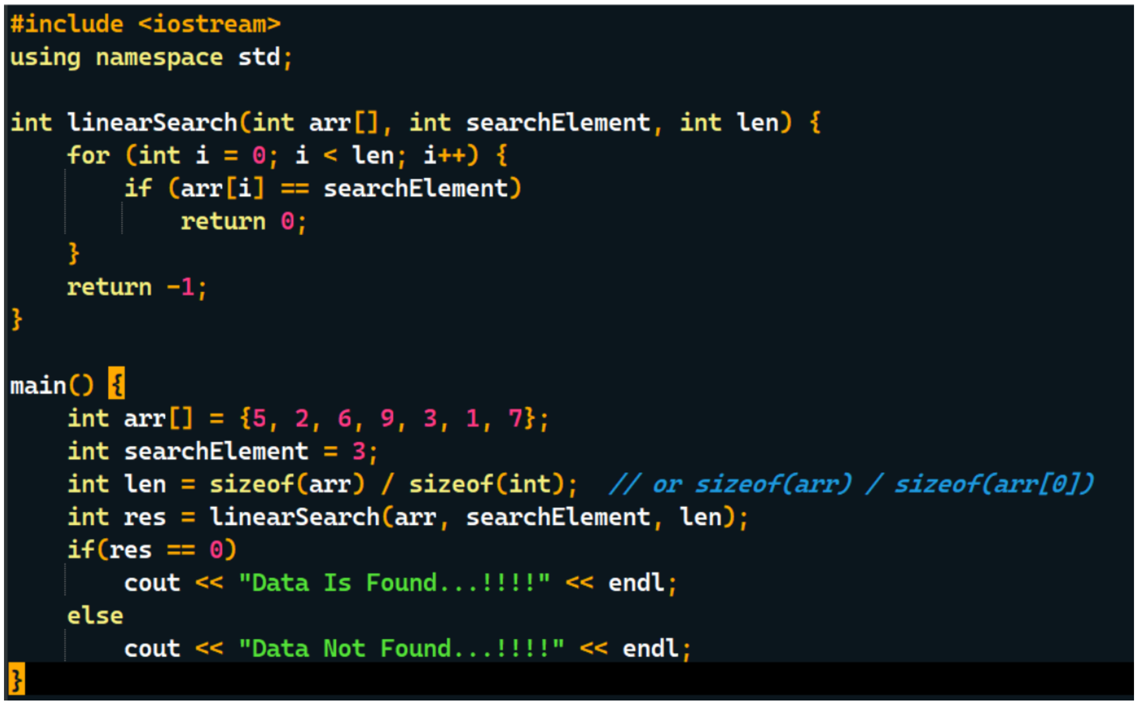
  For finding size of array,

  size_of_array = sizeof(data) / sizeof(int) or sizeof(data) / sizeof(data[0])

  = 24 / 4

  size_of_array = 6

- This above formula we have to write in the main method because if we write the above formula in linearSearch function then at the time of calling function whenever we passed an array that time they only passed pointer not whole array. We need a whole array at a time for calculating the size of the array. Therefore we write the above formula in the main method.



```cpp
#include <iostream>
using namespace std;

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

main() {
    int arr[] = {5, 2, 6, 9, 3, 1, 7};
    int searchElement = 3;
    int len = sizeof(arr) / sizeof(int);  // or sizeof(arr) / sizeof(arr[0])
    int res = linearSearch(arr, searchElement, len);
    if(res == 0)
        cout << "Data Is Found...!!!!" << endl;
    else
        cout << "Data Not Found...!!!!" << endl;
}
```

- In above code, for loop is depends on len variable ie. size of array. Therefore time complexity is not constant. It will depend on the data which the user provides.

  If length of array is 9 then CPU unit time is 9

  If length of array is 3 then CPU unit time is 3

  If length of array is 100000 then CPU unit time is 100000

  Therefore time complexity of code is O(n)

- If we have more data then we need more time for finding search element then we can say the pattern is linear. Therefore we can say time complexity is **O(linear)**.

- We always find time complexity in best case, worst case, average case.
  1) If we find search element at first time means at first index then we can say we are lucky ie. **best case and time complexity is O(1)**.
  2) But if we are not lucky **ie. worst case** and we will traverse the entire array then find the search element at last then **time complexity is O(n)**.