

Linear Search

This C++ program performs a linear search in an array to find a target element. A linear search, also known as sequential search, is a simple searching algorithm that checks every element in the array one by one until the target element is found or until all elements have been examined.

Recursive function to perform linear search in an array

1. The program starts by including the necessary header file for input/output.
2. The **linearSearch** function is defined, which takes an integer array **arr**, the current **index** to start the search, the **size** of the array, and the **target** element to search.
3. Inside the **linearSearch** function, there are three cases:
 - Base Case 1: If the **size** of the array is 0, it means the array is empty, and the target element is not found. In this case, the function returns -1.
 - Base Case 2: If the element at the current **index** matches the **target** element, the function returns the current **index**, indicating that the target element is found.
 - Recursive Case: If the target element is not found at the current **index**, the function makes a recursive call to **linearSearch** with the next index ($\text{index} + 1$) and reduces the size of the array ($\text{size} - 1$). This way, it continues the search in the rest of the array.

Time Complexity:

The time complexity of the linear search algorithm is $O(n)$, where n is the size of the array. In the worst-case scenario, the algorithm may need to check every element in the array to find the target element or determine that it is not present.

Space Complexity:

The space complexity of the program is $O(n)$. This is because it uses an array of size size to store the user-input elements, and the **linearSearch** function makes recursive calls on the call stack, potentially consuming additional space proportional to the size of the array. However, the space used for the recursive call stack is relatively small compared to the input array's size.

Recursive call stack for the approach:

① Linear Search function recursive call tree.
arr = [4, 7, 1, 2, 7, 16] , target = 2, size = 6
linearSearch([4, 7, 1, 2, 7, 16], 0, 6, 2)
├─ linearSearch([4, 7, 1, 2, 7, 16], 1, 6, 2)
│ ├─ linearSearch([4, 7, 1, 2, 7, 16], 2, 6, 2)
│ │ ├─ linearSearch([4, 7, 1, 2, 7, 16], 3, 6, 2)
│ │ │ └─ return 3 (index)