Linear Search Using Recursion in C

Linear Search is a straightforward algorithm used to find a particular element in a list. It sequentially checks each element of the list until a mat

Steps of the Algorithm

1. **Base Case**: If the array size is zero, the element is not found.
2. **Check Current Element**: Compare the target element with the current element.
   - If they match, return the index of the current element.
   - If they do not match, perform the search on the remaining elements.

Pseudocode

C Copy

```c
int linearSearchRecursive(int arr[], int size, int target) {
    // Base case: If size is zero, element is not found
    if (size == 0) {
        return -1;
    }

    // Check current element
    if (arr[size - 1] == target) {
        return size - 1;
    }

    // Search in the remaining array
    return linearSearchRecursive(arr, size - 1, target);
}
```

Source Code in C

Here's the source code for linear search using recursion in C:

C Copy

```c
#include <stdio.h>

// Function to perform linear search using recursion
int linearSearchRecursive(int arr[], int size, int target) {
    // Base case: If size is zero, element is not found
    if (size == 0) {
        return -1;
    }

    // Check current element
    if (arr[size - 1] == target) {
        return size - 1;
    }

    // Search in the remaining array
    return linearSearchRecursive(arr, size - 1, target);
}

int main() {
    int arr[] = {4, 2, 5, 1, 3};
    int size = sizeof(arr) / sizeof(arr[0]);
    int target = 5;
    int result = linearSearchRecursive(arr, size, target);

    if (result == -1) {
        printf("Element not found in the array.\n");
    } else {
        printf("Element found at index: %d\n", result);
    }

    return 0;
}
```

Explanation

1. **Function Definition**: `linearSearchRecursive` is defined to take the array, its size, and the target element as parameters.
2. **Base Case**: If the size of the array is zero, it returns `-1`, indicating the element is not found.
3. **Current Element Check**: It compares the target element with the last element of the array (`arr[size - 1]`).
4. **Recursive Call**: If the current element does not match, the function calls itself with the array size reduced by one, effectively searching the r