11. IMPLEMENTATION OF LINEAR SEARCH

Preamble

Finding pieces within a collection is a typical task in the world of programming. The linear search is one of the most elementary and basic search methods. The specifics of the linear search will be covered in this blog post, along with its implementation in the C programming language, syntax examples, and samples of the expected result.

The fundamental search technique known as linear search that scans every element in a collection until the required element is discovered or the collection has been thoroughly searched. It is often referred to as sequential search. When dealing with short or unordered lists, this algorithm is especially helpful.

Steps

- Starting from the collection's commencement.
- Compared with the target element, compare the current element.
- Return the element's index or location if a match is discovered.
- Return the appropriate indication (e.g., -1) if the collection's conclusion is reached without a match

Implementation in C

```
#include <stdio.h>

int linearSearch(int arr[], int n, int target)

{

      int i;

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

      {

            if (arr[i] == target)

            {

                  return i;  // Element found at index i

            }

    }

    return -1;  // Element not found

}
```

```
int main()
{
      int arr[] = {10, 2, 8, 5, 17};
      int n = sizeof(arr) / sizeof(arr[0]);
      int target = 8;
      int result = linearSearch(arr, n, target);
      if (result == -1)
      {
            printf("Element not found in the array.\n");
    }
      else
      {
            printf("Element found at index: %d\n", result);
      }
      return 0;
}
```

Sample Input and Output

Assuming that the target element is 8, let's use the example array [10, 2, 8, 5, 17]. Running the code yields the following result:

```
Element found at index: 2
```

The element '8' was found in this instance by the linear search technique at index 2 of the array.