DAILY DSA | DAY-18 | Searching algorithms – Linear search| -GOPALKRISHNA A

Searching is something we do every day. Whether they're word searching in documents and databases or repos, the applications for search are nearly endless.

Today we will start with fundamental search algorithms...

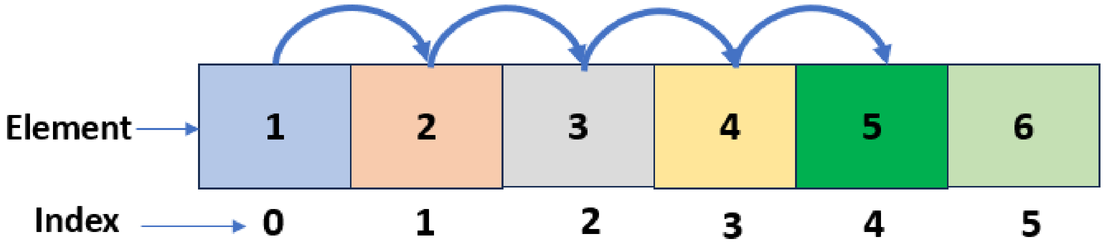
Searching?
Searching is the process of fetching a specific element in a collection of elements. The collection can be an array or a linked list. If an element is found in the list, the process is considered successful, and it returns the location of that element.

Types of Search Algorithms:

- Linear or Sequential search
- Binary search

Linear search: Linear search, often known as sequential search, is the most basic search technique. In this type of search, we **go through the entire list and try to fetch a match for a single element**. If you find a match, then the address of the matching target element is returned otherwise if the element is not found, then it returns a NULL value

INPUT:
Array = [1,2,3,4,5,6]
Search element: 5



Steps to implement linear search:

1. First, read the search element in the array
2. Compare the search element with the first element in the array
3. if both matched, return index & element and terminate linear search
4. if not matched, compare with the next element in the array
5. repeat steps 3 & 4 until the search element is compared with the last element of the array.
6. If the last element in the list does not match, the linear search function will be terminated and returns NULL

Global linear search:
What if we need to find multiple occurrences of an element?

Example:
INPUT: search element: 5, Array = [1,2,3,4,5,6,5,7,8,9,5]
This array has 3 occurrences of 5s we return the indexes where they are in the array this is called global linear search. When we find an index element that matches the search element, the counter will be incremented else search will continue to move on to the next element in the array until the end of the array/list.

Why linear search is not efficient?
There is no doubt that linear search is simple. But because it compares each element one by one, it is time-consuming and therefore not very efficient. If we have to find a number from, say, 100,000 numbers that number is at the last position, a linear search would become quite tedious.

Quick sort, heap sort algorithms we covered in Day-1-3 will be handy to sort.

The time complexity of the Linear search algorithm

- Best case complexity
  - The element being searched could be found in the first position
  - In this case, the search ends with a single successful comparison
  - Thus, in the best-case scenario, the linear search algorithm performs O(1) operations

- Worst case complexity
  - The element being searched may be at the last position in the array or not at all
  - Thus, In the worst-case scenario, the linear search algorithm performs O(n) operations.

- Average case complexity
  - When the element to be searched is in the middle of the array, the average case of the linear search algorithm is O(n)

There is an interesting research paper published on time consumption of linear search
https://www.researchgate.net/publication/284494241_Two_way_Linear_Search_Algorithm

Applications:

- Linear search is easy to implement and effective when the array contains only a few elements
- Linear Search is also efficient when the search is performed to fetch a single search in an unordered list.