Searching Techniques

Introduction to Searching Algorithms

Not even a single day pass, when we do not have to search for something in our day to day life, car keys, books, pen, mobile charger and what not. Same is the life of a computer, there is so much data stored in it, that whenever a user asks for some data, computer has to search it's memory to look for the data and make it available to the user. And the computer has it's own techniques to search through it's memory fast, which you can learn more about in our Operating System tutorial (/operating-system/) series. What if you have to write a program to search a given number in an array? How will you do it? Well, to search an element in a given array, there are four popular algorithms available:

1. Sequential Search.

2. Binary Search.

3.Interpolation search.

4.Jump search.

Sequential Search

Sequential search is a very basic and simple search algorithm. In Sequential search, we search an element or value in a given array by traversing the array from the starting, till the desired element or value is found. It compares the element to be searched with all the elements present in the array and when the element is matched successfully, it returns the index of the element in the array, else it returns -1. Sequential Search is applied on unsorted or unordered lists, when there are fewer elements in a list.

Features of Sequential Search Algorithm

1. It is used for unsorted and unordered small list of elements.

2. It has a time complexity of O(n), which means the time is sequentially dependent on the number of elements, which is not bad, but not that good too.

3. It has a very simple implementation.

Implementing Sequential Search

Following are the steps of implementation that we will be following:

1. Traverse the array using a for loop.

2. In every iteration, compare the target value with the current value of the array. If the values match, return the current index of the array. If the values do not match, move on to the next array element.

3. If no match is found, return -1.

Algorithm for sequential search

for (each item in list)

{

compare search term to current item

if match, save index of matching item break

}

return index of matching item,

or -1 if item not found

```
/* below we have implemented a simple function  for sequential search
in C++
 values[] = array with all the values
  target = value to be found
 n = total number of elements in the array */
int sequentialSearch(int values[], int target, int n)
{
for(int i = 0; i < n; i++)
 {
 if (values[i] == target)
      {
           return i;
      }
   }
return -1;
}
```

Some Examples with Inputs

input: values[] = {5, 34, 65, 12, 77, 35}

target = 77

Output: 4

Input: values[] = {101, 392, 1, 54, 32, 22, 90, 93}

target = 200

Output: -1 (not found)

HomeWorks

- If we have the array [23,17,5,90,12,44,38,84,77] ,search **Sequentially** to find:

1. The value (12).
2. The value (100).
3. The value (23).