ASSIGNMENT ON ARRAYS USING 1) BUBBLE SORT 2) LINEAR SEARCH 3) BINARY SEARCH

- **BY**
- **CH. PAVAN KUMAR**

BUBBLE SORT USING ARRAYS

Bubble sort is a simple sorting algorithm. This sorting algorithm is a comparison-based algorithm in which each pair of elements is compared and the elements are swapped if they are not in order.

The Bubble Sort works by iterating down an array to be sorted from the first element to the last, comparing each pair of elements and switching their positions if necessary. This process is repeated as many times as necessary, until the array is sorted.

- **For Example:** Let us consider int has 5 elements
- int[] arr = { 78, 55, 45, 98, 13 };

- Now, let us perform Bubble Sort.
- Start with the first two elements 78 and 55. 55 is smaller than 78, so swap both of them,
- **STEP1**: The position of 55 and 78 will be changed as shown below.
- {55, 78,45,98, 13}:
- **STEP2**: Now 45 is less than 78, Swapping will be happened.
- {55, 45, 78, 98, 13}:
- **STEP3:** Now 98 is greater than 78, So, no swapping will happen. The positions will remains same.
- **STEP4**: Now swapping happens to 98 and 13. Since 98 is less than 13 swapping will happen. This was the first iteration. After performing all the iterations, we will get our sorted array using Bubble Sort.
- {13, 45, 55, 78, 93}

CODE:

```
 using System;
namespace BubbleSort
{
    /// <summary>
    /// BUBBLE SORT USING ARRAY
    /// </summary>
    //DONE BY: PAVAN//

    class MySort
    {
        static void Main(string[] args)
        {
            int[] data = { 78, 55, 45, 98, 13 };
            int temp;
      {
for (int j = 0; j <= data.Length - 2; j++)
```

```
            {
                if (data[i] > data[i + 1])
                {
                    temp = data[i + 1];
                    data[i + 1] = data[i];
                    data[i] = temp;
                }
            }
        }
        Console.WriteLine("******* OUTPUT IS *******");
        Console.WriteLine("Sorted:");
        foreach (int d in data)
            Console.Write(d + " ");
        Console.ReadLine();
    }
  }
}
```

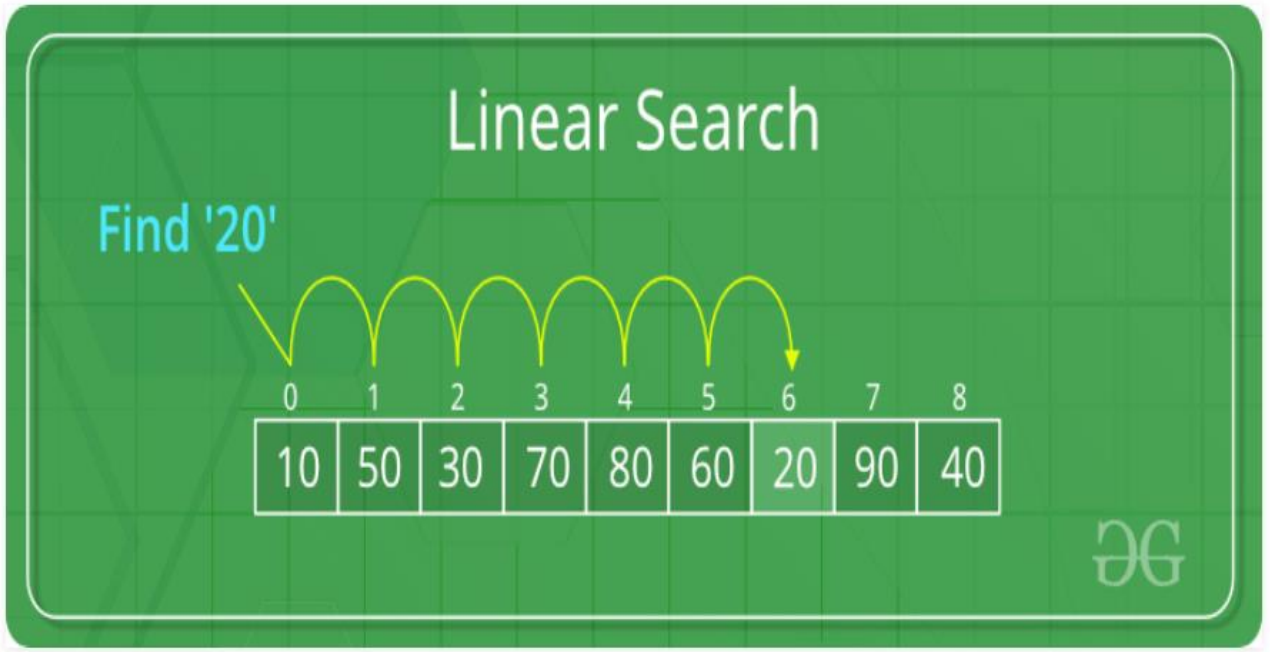
OUTPUT:

LINEAR SEARCH USING ARRAYS

- The Linear search is a technique which allows user to search a particular value from a list of values available in an array.
- The searching starts from the beginning of the array.

LINEAR SEARCH

- The Linear search searches the value which we have given as input in the data in a step by step process to print the output.
- In case we didn't find the output the searching stops I.e.. The search will ends.
  - Start from the leftmost element of input and one by one compare with each element of input



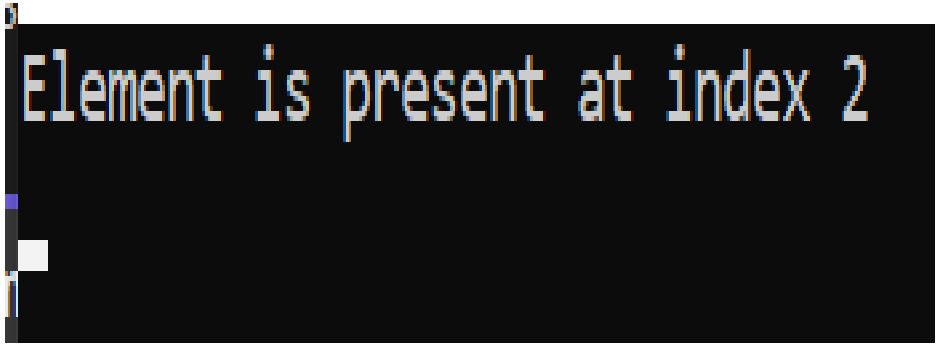
CODE:

```
using System;
using System. Collections;
using System. Text;
using System. Linq;
namespace Linear
{
    /// <summary>
    /// LINEAR SEARCH USING ARRAYS
    /// DONE BY: PAVAN
    /// </summary>

class Linear Search
{
    public static int Search(int[] data, int x)
    {

        for (int i = 0; i < data. Length; i++)
        {
            if (data[i] == x)
                return i;
```

```
    }
        return -1;
     }
public static void Main()
        {
            int[] data = { 5, 10, 15, 20, 25 };
            int a = 15;
            int result = Search(data, a);
            if (result == -1)
                Console.WriteLine("Element is not present in array");
            else
                Console.WriteLine("Element is present at index "+result);
            Console.ReadLine();
        }
    }
}
```



OUTPUT:

BINARY SEARCH USING ARRAY

- Binary Search is a method to find the required element in a sorted array by repeatedly halving the array and searching in the half.
- This method is done by starting with the whole array. Then it is halved. If the required data value is greater than the element at the middle of the array, then the upper half of the array is considered.
- The idea of binary search is to use the information that the array is sorted and reduce the time complexity.

THANK YOU