Homework2Q2

What is the Big-O Time Complexity Analysis of Linear Search? - LC

- Process
  - Step 1: Please use Loop Analysis method to analyze

    public static int search(int arr[], int x)

    Please explain your answer.

```
// Java code for linearly search x in arr[]. If x
// is present then return its location, otherwise
// return -1

class GFG
{
public static int search(int arr[], int x)
{
    int n = arr.length;
    for(int i = 0; i < n; i++) =>O(n)
    {
        if(arr[i] == x)
            return i;
    }
    return -1;
}

public static void main(String args[])
{
    int arr[] = { 2, 3, 4, 10, 40 };
    int x = 10;

    int result = search(arr, x);
    if(result == -1)
        System.out.print("Element is not present in
array");
    else
        System.out.print("Element is present at index " +
result);
}
}
```

Ans-In the given code, there is a single loop that iterates through the array "arr" from the first element to the last. The loop variable "i" starts at 0 and increments by 1 until it reaches the length of the array.

Let's denote the length of the array as "n". The loop iterates "n" times, checking each element of the array until it finds the target element or reaches the end of the array. Therefore, the time complexity of the linear search algorithm can be expressed as O(n).