Question-2

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++)
    {
        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);
}
}
```

I have a single loop that linearly iterates over the array elements. The loop runs from the first element to the last element, checking if each element matches the target element (x).

The number of iterations in the loop depends on the size of the array (n). In the worst case, the loop will go through all n elements if the target element is not found.

Therefore, the time complexity of the linear search algorithm is O(n). where n is the number of elements in the array.