Week -2 Homework-2

Name: Himavanth reddy putta Id:18759

Q29) What is the Big-O Time Complexity Analysis of Linear Search? - LC

* Process
  + Step 1: Please use [Loop Analysis](https://hc.labnet.sfbu.edu/~henry/npu/classes/algorithm/geeksforgeeks/slide/analyze_loop.html) 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);

}

}

Big O = O(n)

My explanation:

->The loop begins with the loop counter 'i' to 0 known as O(1), is required for this step.

-> The loop counter 'i' is increased by 1 after each iteration.

->As a result, the linear search algorithm's complexity using the Loop Analysis is O(n).