DAA Assignment 4

IIT2019087 Gautum Kumar
IIT2019088 Ritik Kumar
IIT201989 Pranav Sai

March 12, 2021

Table of contents

Question

Given a sorted array and a value x, the floor of x is the largest element in array smaller than or equal to x. Write efficient functions to find floor of x.

floor

Floor :

It gives the greatest integer less than or equal the given value.

Algos

Algorithms

Here we are following two algorithms to find the floor of x in the given array.

- Linear Approach
- Binary Approach

Linear Search

Concept

We are simply finding first greatest value of x by traversing the array using for loop

- **If the current element is greater than x then return the just previous value of the current element.**
- If there is no number greater than x then print the last element
- If the first number is greater than x then print -1

Time CompleXity will be O(N).

Code

Algorithm

```
int a[1000];

int fuc(int x,int n){

 if(x<a[0]) return -1;
 if(x>=a[n-1])return a[n-1];

    for(int i=0;i<n;i++){

                if(a[i]>x)
                 return a[i-1];
    }
}
```

Graph Analysis

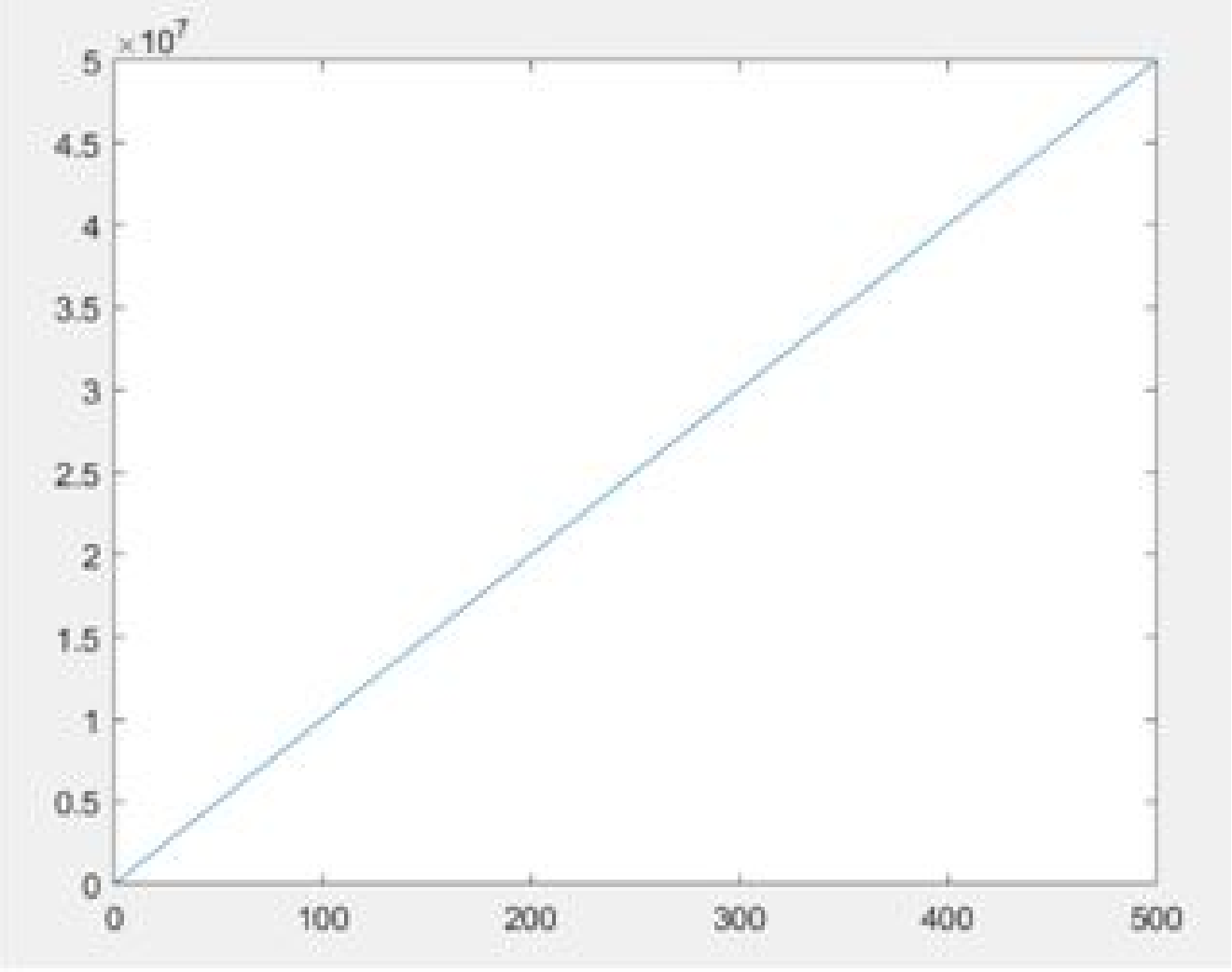

Figure 1: Naïve Approach

Binary Search

Concept

The idea is to use Binary Search Approach to find the floor of a number x in a sorted array by comparing it to the middle element and dividing the search space into half will be more efficient than Linear Approach.

- create three variables low = 0, mid and high = n-1 and Run a while loop until and unless low is less than high.
- **check if the middle ((low + high) /2) element is less than x, if yes then update the low, i.e low = mid + 1, and update answer with the middle element. In this step we are reducing the search space to half. Else update the low, i.e high = mid – 1.**
- If x is greater than or equal to mid-1 element and less than mid element in the array return element of mid -1.

Time CompleXity will be O(log(N)).

Code

Algorithm

```
int fuc(int x,int low,int high){
int mid;
 while(low<high){

    mid=(low+high)/2;

   if(x==a[mid]) return a[mid];
   if(x<a[mid]) high=mid-1;
   else low=mid+1;
    }

   if(a[mid-1]<=x&&x<a[mid])
   return a[mid-1];
```

Graph Analysis

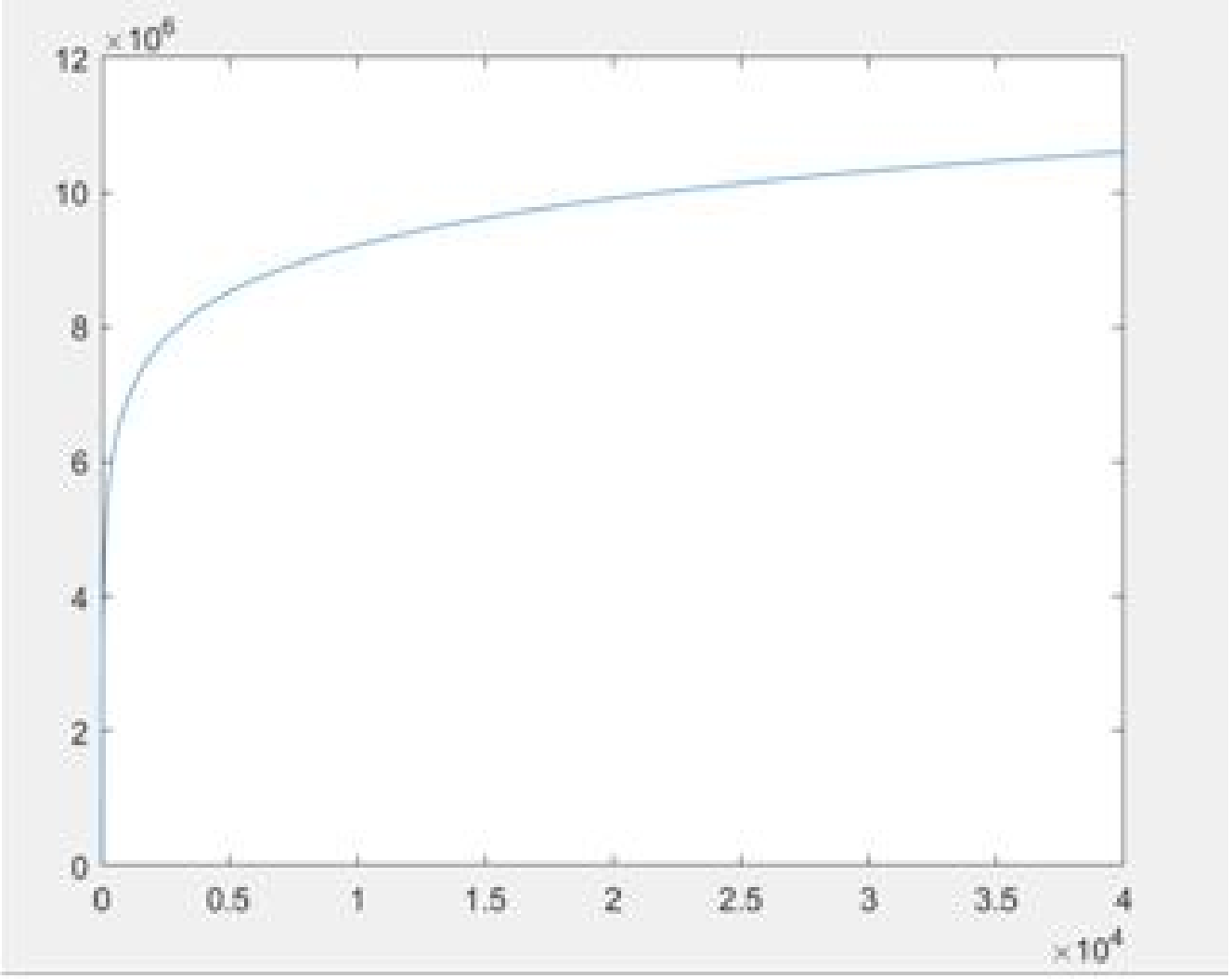

Figure 2: Binary Search

Combined Graph Analysis

Linear and Binary both

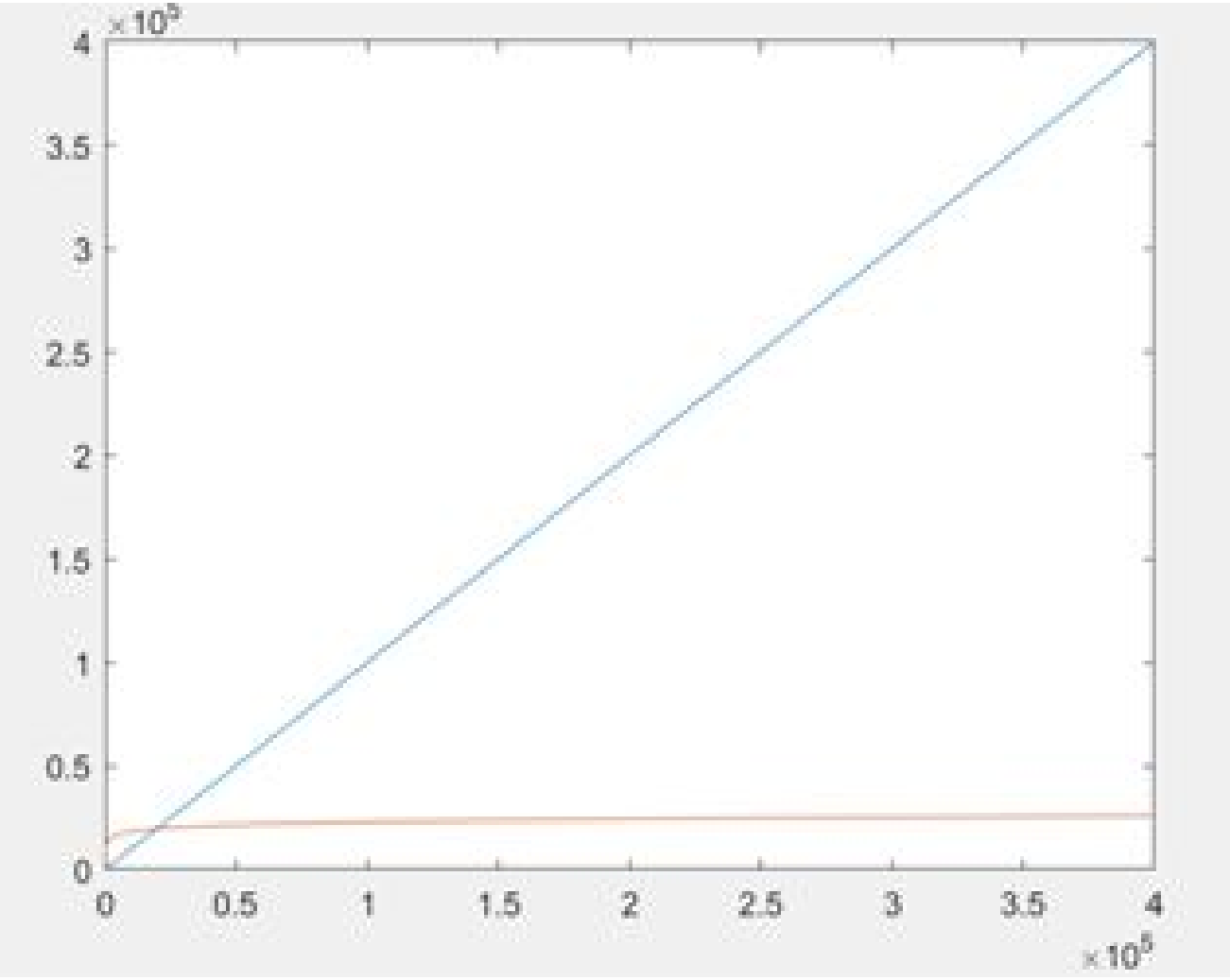

Figure 3: Comparison between naïve and binary search approach

Conclusion

Summary

Here we've seen two approaches to find floor of x by Linear and Binary Approach. Binary Approach is undoubtedly more efficient aglo in this case.