

GeeksforGeeks

A computer science portal for geeks

GeeksQuiz

- [Home](#)
- [Algorithms](#)
- [DS](#)
- [GATE](#)
- [Interview Corner](#)
- [Q&A](#)
- [C](#)
- [C++](#)
- [Java](#)
- [Books](#)
- [Contribute](#)
- [Ask a Q](#)
- [About](#)

[Array](#)

[Bit Magic](#)

[C/C++](#)

[Articles](#)

[GFactS](#)

[Linked List](#)

[MCQ](#)

[Misc](#)

[Output](#)

[String](#)

[Tree](#)

[Graph](#)

Search in an almost sorted array

Given an array which is sorted, but after sorting some elements are moved to either of the adjacent positions, i.e., $arr[i]$ may be present at $arr[i+1]$ or $arr[i-1]$. Write an efficient function to search an element in this array. Basically the element $arr[i]$ can only be swapped with either $arr[i+1]$ or $arr[i-1]$.

For example consider the array $\{2, 3, 10, 4, 40\}$, 4 is moved to next position and 10 is moved to previous position.

Example:

Input: $arr[] = \{10, 3, 40, 20, 50, 80, 70\}$, $key = 40$

Output: 2

Output is index of 40 in given array

Input: arr[] = {10, 3, 40, 20, 50, 80, 70}, key = 90

Output: -1

-1 is returned to indicate element is not present

A simple solution is to linearly search the given key in given array. Time complexity of this solution is $O(n)$. We can modify [binary search](#) to do it in $O(\log n)$ time.

The idea is to compare the key with middle 3 elements, if present then return the index. If not present, then compare the key with middle element to decide whether to go in left half or right half. Comparing with middle element is enough as all the elements after mid+2 must be greater than element mid and all elements before mid-2 must be smaller than mid element.

Following is C++ implementation of this approach.

```
// C++ program to find an element in an almost sorted array
#include <stdio.h>

// A recursive binary search based function. It returns index of x in
// given array arr[l..r] is present, otherwise -1
int binarySearch(int arr[], int l, int r, int x)
{
    if (r >= l)
    {
        int mid = l + (r - l)/2;

        // If the element is present at one of the middle 3 positions
        if (arr[mid] == x) return mid;
        if (mid > l && arr[mid-1] == x) return (mid - 1);
        if (mid < r && arr[mid+1] == x) return (mid + 1);

        // If element is smaller than mid, then it can only be present
        // in left subarray
        if (arr[mid] > x) return binarySearch(arr, l, mid-2, x);

        // Else the element can only be present in right subarray
        return binarySearch(arr, mid+2, r, x);
    }

    // We reach here when element is not present in array
    return -1;
}

// Driver program to test above function
int main(void)
{
    int arr[] = {3, 2, 10, 4, 40};
    int n = sizeof(arr)/ sizeof(arr[0]);
    int x = 4;
    int result = binarySearch(arr, 0, n-1, x);
    (result == -1)? printf("Element is not present in array")
                  : printf("Element is present at index %d", result);

    return 0;
}
```

Output:

Element is present at index 3

Time complexity of the above function is $O(\log n)$.

This article is contributed by **Abhishek**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Related Topics:

- [Find Union and Intersection of two unsorted arrays](#)
- [Pythagorean Triplet in an array](#)
- [Maximum profit by buying and selling a share at most twice](#)
- [Design a data structure that supports insert, delete, search and getRandom in constant time](#)
- [Print missing elements that lie in range 0 – 99](#)
- [Iterative Merge Sort](#)
- [Group multiple occurrence of array elements ordered by first occurrence](#)
- [Given a sorted and rotated array, find if there is a pair with a given sum](#)



Tweet

3

g+1

1

Writing code in comment? Please use ideone.com and share the link here.

42 Comments

GeeksforGeeks

1 Login ▾

♥ Recommend 1

🔗 Share

Sort by Newest ▾



Join the discussion...



wxyz • 4 months ago

I think elements positioned at odd/even indices are sorted. Can't we run binary search on these two sequences.

^ | ▾ • Reply • Share ›



vicharak • 5 months ago

@GeeksforGeeks instead of comparing middle three elements everytime can we compare only three and go like this.... this will be helpful in case when there are too many elements

```

int binarySearch(int arr[], int l, int r, int x)
{
    if (r >= l)
    {
        int mid = l + (r - l)/2;

        if (arr[mid] == x)

```

```
return mid;
elseif(mid > l && arr[mid] < x )
{
if (arr[mid+1] == x)
return (mid + 1);
else
return binarySearch(arr, l, mid-1, x);
}
```

[see more](#)

^ | v • Reply • Share ›



vicharak → vicharak • 5 months ago

please anybody give your opinion if my way is bit more time complex just check wether it works fine or not

^ | v • Reply • Share ›



Random • 7 months ago

Above code is perfectly fine and handle all the cases.

^ | v • Reply • Share ›



Ankit Jain • 7 months ago

this code won't work for 3,4,5,6,7

correection can be found here <http://sprunge.us/iCNg>

1 ^ | v • Reply • Share ›



Siya → Ankit Jain • 5 months ago

I have checked its working fine.

^ | v • Reply • Share ›



Joy Dutta • 8 months ago

<http://ideone.com/3edRwz>

^ | v • Reply • Share ›



munjal • 8 months ago

<http://geekforgeekss.blogspot....>

^ | v • Reply • Share ›



Anil Kumar V • 8 months ago

Do we still require mid > l, mid < r conditions?

^ | v • Reply • Share ›



Rathin → Anil Kumar V • 8 months ago



randomguy202 • 9 months ago

Yep, just in case for corner values. in the case of looking for 1 in 1,2,4,3.

^ | v • Reply • Share ›



Rashid Khan • 9 months ago

what if we can search for the culprit element and then swap with $\text{arr}[i-1]$ or $\text{arr}[i+1]$ accordingly and then the problem becomes simple to search using binary search

^ | v • Reply • Share ›



Times Square → Rashid Khan • 9 months ago

How would you search the culprit in the first place when the array is partially sorted?

1 ^ | v • Reply • Share ›



khan → Times Square • 8 months ago

in one scan through the array we can make the array sorted and then apply binary search

in the first scan compare each element to its next if next element is smaller than previous swap these two similarly in one scan we can make the array sorted and then apply binary search

^ | v • Reply • Share ›



randomguy202 → Khan • 4 months ago

Your solution will also work but it will be linear time ... $O(n)$. Using the solution given by G4G will be $O(\log n)$ hence preferred.

1 ^ | v • Reply • Share ›



rohit.raj10 → Times Square • 9 months ago

Best possible will be quicksort and if almost sorted then it will be close to best case with very few swapping. Modified binary sort will be better which will be in $O(\log N)$ while in this case Sorting will be $(n \log n)$.

^ | v • Reply • Share ›



Mission Peace • 9 months ago

Checkout lots of questions done in Java

<https://github.com/mission-pea...>

^ | v • Reply • Share ›



Manish Sharma • 9 months ago

It won't work if there are only two element in the array and searched for a non-existent key(it may throw a non-existent key error)

^ | v • Reply • Share ›



Naveed Anjum • 9 months ago



Here is the solution in JAVA

```
public class FindTheKeyInAlmostSortedArray {  
  
    /**  
  
    * @param args  
  
    */  
  
    public static void main(String[] args) {  
  
        // TODO Auto-generated method stub  
  
        int arr[] = {10, 3, 40, 20, 50, 80, 70};  
  
        int searchKey =3;  
  
        System.out.println(binarySearch(arr,0,arr.length,arr.length/2,searchKey));  
  
    }  
}
```

[see more](#)

^ | v • Reply • Share ›



zealfire • 9 months ago

what will happen in this case :Input: arr[] = {10, 3, 40, 50, 80, 20, 70}, key = 20.I am confused becoz if this case is possible then above solution would not work.Please clarify?

^ | v • Reply • Share ›



Sumedh → zealfire • 9 months ago

It is not an almost sorted array.

^ | v • Reply • Share ›



zealfire → Sumedh • 9 months ago

the problem doesn't clarify is this transformation possible

initial array:3,10,20,40,50,70,80

10,3,20,40,50,70,80

10,3,40,20,50,70,80

10,3,40,50,20,70,80

10,3,40,50,20,80,70

final array after transformation:10,3,40,50,80,20,70

^ | v • Reply • Share ›



randomguy202 → zealfire • 4 months ago

Nope, 20 has moved two places away from original position.

^ | v • Reply • Share ›

**Mission Peace** • 9 months ago

Most of geeks for geeks solutions written in java

<https://github.com/mission-pea...>

1 ^ | v • Reply • Share ›

**RK** • 9 months ago

```
def search(arr, sidx, eid, key)
while sidx <= eid
return sidx if arr[sidx] == key
sidx += 1
end
end

def binary_search(arr, sidx, eid, key)
return search(arr, sidx, eid, key) if eid - sidx <= 3
mid = (sidx + eid)/2
if arr[mid] >= key
return binary_search(arr, sidx, mid + 1, key)
else
return binary_search(arr, mid - 1, eid, key)
end
end
```

```
a = [1,0,2,3,5,4,7,6,8,9]
```

```
puts binary_search(a, 0, a.length, 4) ==> 5
```

^ | v • Reply • Share ›

**GeeksforGeeks** Mod • 9 months ago

@All, thanks for your inputs. We have updated the code to compare with middle element only. Also, the two recursive calls are optimized to be called from l to mid-2 and mid+2 to h. Please let us know if you find anything incorrect in the code.

1 ^ | v • Reply • Share ›

**Just-a-Beginner** • 9 months ago

Can anyone explain the intuition behind finding second maxm.

^ | v • Reply • Share ›

**augu** • 9 months ago

why dont we pass mid-1,mid,mid+1 to findsecond...?

^ | v • Reply • Share ›

**Guest** • 9 months ago

**@GeeksforGeeks**

Suppose we have an array not having unique numbers then this fails!

Suppose we have an array={1,2,2,2,3};

and we search for 1 in it..the result is -1! which is clearly wrong!

^ | v • Reply • Share ›

**GeeksforGeeks** Mod → Guest • 9 months ago

This seems to be working for this case. Please see <http://ideone.com/ndYiNP>

1 ^ | v • Reply • Share ›

**Anurag Singh** • 9 months ago

I'm not getting why we need "findSecond" method here. If key (x) doesn't match with mid, mid-1 or mid + 1, then why can't we just call `binarySearch(arr, l, mid-1, x)` OR `binarySearch(arr, mid+1, r, x)`. Can someone give an example input array where search will fail without having "findSecond" ?

^ | v • Reply • Share ›

**Kartik** → Anurag Singh • 9 months ago

How would you decide which of the two to call?

^ | v • Reply • Share ›

**Anurag Singh** → Kartik • 9 months ago

if `arr[mid] > x` then `binarySearch(arr, l, mid-1, x)` else `binarySearch(arr, mid+1, r, x)`.

^ | v • Reply • Share ›

**mak** → Anurag Singh • 9 months ago

it wont solve... see this.

3,2,10,4,40.`arr[mid]`(which is greater than 4) then u ll search in 3,2.

^ | v • Reply • Share ›

**Anurag Singh** → mak • 9 months ago

The matching of mid-1, mid and mid+1 will take care of that.

^ | v • Reply • Share ›

**Kartik** → Anurag Singh • 9 months ago

Thanks Anurag, it looks good to me.

Abhishek?

^ | v • Reply • Share ›

**ritika** • 9 months ago

i think the last return statement in `findSecond` is unnecessary and it should be l in third.

^ | v • Reply • Share ›



Kartik → ritika • 9 months ago

It is needed. You can understand the need of this by drawing recursion tree for a sorted array like {1, 5, 10} and searching an element like 20.

^ | v • Reply • Share ›



Guru Singe • 9 months ago

if (arr[l] <= arr[h] && arr[l] >= arr[mid]) return l;

^ | v • Reply • Share ›



Guest • 9 months ago

In findSecond function, the third if should return l instead of h.

3 ^ | v • Reply • Share ›



Sonia Gandhi → Guest • 9 months ago

bas kar beta kitna pelega! election ke liye tayyari kar..

10 ^ | v • Reply • Share ›



Just-a-Beginner → Sonia Gandhi • 9 months ago

LMFAO!!

1 ^ | v • Reply • Share ›



GeeksforGeeks Mod → Guest • 9 months ago

Thanks for pointing this out. We have corrected the code.

^ | v • Reply • Share ›

[Subscribe](#)

[Add Disqus to your site](#)

[Privacy](#)

-
-
-
- - [Interview Experiences](#)
 - [Advanced Data Structures](#)
 - [Dynamic Programming](#)
 - [Greedy Algorithms](#)
 - [Backtracking](#)
 - [Pattern Searching](#)
 - [Divide & Conquer](#)
 - [Mathematical Algorithms](#)
 - [Recursion](#)
 - [Geometric Algorithms](#)
-

• Popular Posts

- [All permutations of a given string](#)
- [Memory Layout of C Programs](#)
- [Understanding “extern” keyword in C](#)
- [Median of two sorted arrays](#)
- [Tree traversal without recursion and without stack!](#)
- [Structure Member Alignment, Padding and Data Packing](#)
- [Intersection point of two Linked Lists](#)
- [Lowest Common Ancestor in a BST.](#)
- [Check if a binary tree is BST or not](#)
- [Sorted Linked List to Balanced BST](#)

Follow @GeeksforGeeks

• Recent Comments

- It_k
i need help for coding this function in java...
[Java Programming Language](#) · [1 hour ago](#)
- Piyush
What is the purpose of else if (recStack[*i])...
[Detect Cycle in a Directed Graph](#) · [1 hour ago](#)

- [Andy Toh](#)

My compile-time solution, which agrees with the...

[Dynamic Programming | Set 16 \(Floyd Warshall Algorithm\)](#) · [1 hour ago](#)

- [lucy](#)

because we first fill zero in first col and...

[Dynamic Programming | Set 29 \(Longest Common Substring\)](#) · [2 hours ago](#)

- [lucy](#)

@GeeksforGeeks i don't n know what is this long...

[Dynamic Programming | Set 28 \(Minimum insertions to form a palindrome\)](#) · [2 hours ago](#)

- [manish](#)

Because TAN is not a subsequence of RANT. ANT...

[Given two strings, find if first string is a subsequence of second](#) · [2 hours ago](#)

•

@geeksforgeeks, [Some rights reserved](#) [Contact Us!](#)

Powered by [WordPress](#) & [MooTools](#), customized by geeksforgeeks team