Find k closest elements to a given value

Given a sorted array arr[] and a value X, find the k closest elements to X in arr[].
Examples:

```
Input: K = 4, X = 35
       arr[] = {12, 16, 22, 30, 35, 39, 42,
                45, 48, 50, 53, 55, 56}
Output: 30 39 42 45
```

Note that if the element is present in array, then it should not be in output, only the other closest elements are required.

In the following solutions, it is assumed that all elements of array are distinct.

A **simple solution** is to do linear search for k closest elements.
1) Start from the first element and search for the crossover point (The point before which elements are smaller than or equal to X and after which elements are greater). This step takes O(n) time.
2) Once we find the crossover point, we can compare elements on both sides of crossover point to print k closest elements. This step takes O(k) time.

The time complexity of the above solution is O(n).

An **Optimized Solution** is to find k elements in O(Logn + k) time. The idea is to use Binary Search to find the crossover point. Once we find index of crossover point, we can print k closest elements in O(k) time.

```
#include<stdio.h>

/* Function to find the cross over point (the point before
   which elements are smaller than or equal to x and after
   which greater than x)*/
int findCrossOver(int arr[], int low, int high, int x)
{
  // Base cases
  if (arr[high] <= x) // x is greater than all
    return high;
  if (arr[low] > x)  // x is smaller than all
    return low;

  // Find the middle point
  int mid = (low + high)/2;  /* low + (high - low)/2 */

  /* If x is same as middle element, then return mid */
  if (arr[mid] <= x && arr[mid+1] > x)
    return mid;

  /* If x is greater than arr[mid], then either arr[mid + 1]
    is ceiling of x or ceiling lies in arr[mid+1...high] */
  if(arr[mid] < x)
      return findCrossOver(arr, mid+1, high, x);

  return findCrossOver(arr, low, mid - 1, x);
}

// This function prints k closest elements to x in arr[].
// n is the number of elements in arr[]
void printKclosest(int arr[], int x, int k, int n)
{
    // Find the crossover point
    int l = findCrossOver(arr, 0, n-1, x); // le
    int r = l+1;   // Right index to search
    int count = 0; // To keep track of count of elements already printed

    // If x is present in arr[], then reduce left index
    // Assumption: all elements in arr[] are distinct
    if (arr[l] == x) l--;
```

```
    // Compare elements on left and right of crossover
    // point to find the k closest elements
    while (l >= 0 && r < n && count < k)
    {
        if (x - arr[l] < arr[r] - x)
            printf("%d ", arr[l--]);
        else
            printf("%d ", arr[r++]);
        count++;
    }

    // If there are no more elements on right side, then
    // print left elements
    while (count < k && l >= 0)
        printf("%d ", arr[l--]), count++;

    // If there are no more elements on left side, then
    // print right elements
    while (count < k && r < n)
        printf("%d ", arr[r++]), count++;
}

/* Driver program to check above functions */
int main()
{
   int arr[] ={12, 16, 22, 30, 35, 39, 42,
               45, 48, 50, 53, 55, 56};
   int n = sizeof(arr)/sizeof(arr[0]);
   int x = 35, k = 4;
   printKclosest(arr, x, 4, n);
   return 0;
}
```

Output:

```
39 30 42 45
```

The time complexity of this method is O(Logn + k).

Exercise: Extend the optimized solution to work for duplicates also, i.e., to work for arrays where elements don't have to be distinct.

This article is contributed by **Rahul Jain**. 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

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

50 Comments GeeksforGeeks 1 Login

Recommend Share Sort by Newest

Join the discussion…

Amit • 2 months ago

get sum of K elements before (less than x). Store a running sum and keep sliding one element in each iteration. of K elements (subtract first and add last element) . This we have to do utmost K times. Which ever has sum least distant from (Absolute(sum-K*x) is the correct window

• Reply • Share ›

Abhinav Gupta • 4 months ago

Please find a more elegant and readable code for the above problem:
http://ideone.com/N7YxWi

Please let me know if there are any bugs in the above code.

• Reply • Share ›

Swapnil • 5 months ago

Closest Element in the given example are 30, 39, 42, 45 .
Why not 22 30 39 42 ???

• Reply • Share ›

kAos ➚ Swapnil • 3 months ago

not 22 coz (35 - 22 > 45 - 35)
hence 45 is printed

• Reply • Share ›

m_asif • 6 months ago

http://ideone.com/hlfsIY

• Reply • Share ›

Temujin • 8 months ago

http://ideone.com/L8man8

Uses upper_bound / lower_bound, also tackles for Corner cases !

• Reply • Share ›

rihansh ➦ Temujin • 5 months ago

but surely u will not allow to use inbuilt function during a tech interview. so, its better to make our own :D isn't ??

• Reply • Share ›

Arun Kumaran Sabtharishi • 9 months ago

How about the below solution?

private static int[] getClosest(int num, int element, int arr[]){

int elementIndex = -1;

int closeArray [] = new int[num];

for(int i=0;i<arr.length;i++) {="" if(arr[i]="=element){" elementindex="i;" }="" }="" for(int="" i="0," j="elementIndex-1," k="elementIndex+1;i<num;i++){" int="" diff1="element-arr[j];" int="" diff2="arr[k]-element;" if(diff1<diff2){="" closearray[i]="arr[j];" j--;="" }="" else="" if(diff1="">diff2) {

closeArray[i]=arr[k];

k++;

}

else{

closeArray[i]=arr[j];

if(i<num){ i++;="" closearray[i]="arr[k];" j--;="" k++;="" }="" else="" {="" break;="" }="" }="" return="" closearray;="" }="">

• Reply • Share ›

ajayv • 9 months ago

private static void printKClosest(int[] arr, int n,int goal,int K) {
int index = Arrays.binarySearch(arr, 0, n-1, goal);
int l = index-1;
int r = index+1;
int count = 0;
while(l>=0 && r<n &&="" count="" <="" k--)="" {="" if(goal-arr[l]<="" arr[r]-goal){="" system.out.print(arr[l]+"="" ");="" l--;="" }="" else="" if(goal-arr[l]="">= arr[r]-goal){
System.out.print(arr[r]+" ");
r++;
}
}

j

• Reply • Share ›

ajayv ajayv • 9 months ago
This solution is correct if the X is present is array. And that is the reason why we need to find the cross over point , if X is not present.
• Reply • Share ›

Jun • 10 months ago
How to proceed if, the cross over number is nt present in the array??????????????????????
• Reply • Share ›

DS+Algo=Placement Jun • 9 months ago
Which case r u talking about, crossover point will always present in array
• Reply • Share ›

Jun DS+Algo=Placement • 9 months ago
the case where we need to find k closest elements from a number, which is nt present in the array
• Reply • Share ›

DS+Algo=Placement Jun • 9 months ago
the just lesser term will be returned as crossover point.
For array in above example, if number is given 21, then 16 will be returned as crossover point
• Reply • Share ›

ks • a year ago
what if array is not sorted?
• Reply • Share ›

GOPI GOPINATH ks • a year ago
The problem statement itself says that array is sorted, so we need not bother about this case, Explicitly we always give a sorted array.
• Reply • Share ›

Vivek Garg GOPI GOPINATH • 10 months ago
If it is not sorted then it will be an different problem and we can use only method 1 of this problem and the time complexity will be o(n).
• Reply • Share ›

Mrinal Ahlawat Vivek Garg • 10 months ago
no not even the first method will work in case the array is not sorted
1 • Reply • Share ›

Abhishek ➚ Mrinal Ahlawat • 10 months ago

better to sort array and then do the same operation.
compelxity will b O(2logn+k)

• Reply • Share ›

Mrinal Ahlawat ➚ Abhishek • 10 months ago

you cannot sort an array in logn complexity...if you sort it then proceed the overall complexity will become O((n+1)logn + k) which is equivalent to O(nlogn + k)

• Reply • Share ›

Abhishek ➚ Mrinal Ahlawat • 10 months ago

yes my mistake!!! :)

• Reply • Share ›

ocean ➚ Abhishek • 10 months ago

when the array is not sorted, then we can use the following method:
1.build a heap of first k numbers on the based on the absolute difference between the given value and the number.
2. max-heapfiy this.
3. for all the remaining elements of the array, compare the array elements with the given number. if the absolute difference between the root and the array element is less, then this element as the root and max-heapify the heap.
4. the resulting heap gives the k closest numbers.

1 • Reply • Share ›

Suvodip Bhattacharya • a year ago

```
#include<iostream>
#include<cmath>
using namespace std;

int Binsearch(int arr[],int num,int l,int h)
{
if(arr[(l+h)/2]==num) return (l+h)/2;

int mid=(l+h)/2;
return (arr[mid] < num ) ? Binsearch(arr,num,mid+1,h) : Binsearch(arr,num,l,mid-1) ;
}
int FindIndex(int arr[],int size,int num)
{
return Binsearch(arr,num,0,size-1);
}
```

```
void KclosestElement(int arr[],int size,int k,int num)
{
int index=FindIndex(arr,size,num);
```

see more

• Reply • Share ›

kaushik Lele • a year ago

Optimized solution uses binary search. So I thought that it should be explained in very simple way.
Call binary search to find out position of number. Say, p th position. Then take two counters i = p-1; j = p+1
And then find out the distance from the number; which ever number is less it is added to final list. And corresponding counter is advanced i.e. i-- or j++

But then I realized that calling binary search will not help; if given number is not present in array.
And hence your code to find out crossover point is required.
I thought of mentioning this so that other may no waste time in same thinking.

@Geek if you could highlight this point it will be helpful

3 • Reply • Share ›

Himanshu Dagar ➚ kaushik Lele • a month ago

Even in that case when element is not present , your this approach will work. You have to just start with i=p-1,j=p

where p is the position where that x needs to be inserted.
So no need to take the special concept of crossover point.
Just apply normal binarySearch

• Reply • Share ›

Renjith • a year ago

Java Implementation:

http://ideone.com/8ciYde

• Reply • Share ›

codeconnoisseur ➚ Renjith • a year ago

nice !

• Reply • Share ›

Tejas Joshi • a year ago

We can make it more efficient if we need only begin and end index of k closest values.

We can find it in O(log n + log k).
For the given example the answer would be 3 and 7.
Let say the value is V.
1) We can search the index of V in O(log n) using binary search. Let say it is x.
2) Here we need to find k closest values.
- Compare the value at index (x-(k/2)) (left side) and (x+(k/2)) (right side) with the absolute difference with V.
- After the comparison one of the (k/2) values will be added.
- We will extend (k/4) to the selected portion and reduce (k/4) for the not selected portion and continue until we will get k values. Let say we have considered the portion at right side so the next comparison would be values at (x-(k/4)) and (x+(k/2)+(k/4)).

2 • Reply • Share ›

kaushik Lele ➚ Tejas Joshi • a year ago

What if V is not present in array ?

• Reply • Share ›

Tejas Joshi ➚ kaushik Lele • a year ago

We can take ceil OR floor and set the condition accordingly.

• Reply • Share ›

kaushik Lele ➚ Tejas Joshi • a year ago

You have said "We can search the index of V in O(log n) using binary search" ; if V is not present in array binary search will not give any valid position in array.

1 • Reply • Share ›

Tejas Joshi ➚ kaushik Lele • a year ago

Please refer : http://www.geeksforgeeks.org/s...

1 • Reply • Share ›

Savan Popat ➚ Tejas Joshi • a year ago

What if all the k elements are present in one side of V.
for example:
12 13 14 15 17 24 30 40 50. V is 17 and k = 4
ans should be 12 13 14 15
please reply

• Reply • Share ›

Tejas Joshi ➚ Savan Popat • a year ago

k=4
First we will search for index of 17 --> 4 let say it is x. so x=4.
Initially,
index, diff = k

index_diff = k
left = right = x

Now, (in the loop)
index_diff = index_diff/2 (2)
left = left-index_diff (2)
right = right+index_diff (6)

We will compare the values at x+index_diff and x-index_diff w.r.t. diff
i.e. A[right]-A[x] and A[x]-A[left] ==> 13 > 3
so we added k/2 elements (14,15)

Next iteration,
index_diff = index_diff/2 (1)
left = left-index_diff (1)
right = right-index_diff (5)

Now we will compare again A[right]-A[x] and A[x]-A[left] ==> 7 > 4
so we have added k/4 elements. (13)

and continue until we include k elements.
This will take O(log k) as we need only the indexes.

• Reply • Share ›

Guest • a year ago

#include <stdio.h>

#define MAXSIZE 100

int main()

{

int arr[MAXSIZE];

int n,i,x,p,k,c=0;

printf("No of elements of array are :");

scanf("%d",&n);

printf("\nEnter the elements of array\n");

for(i=0;i<n;i++) scanf("%d",&arr[i]);="" printf("\nenter="" the="" value="" of="" x(element)="" and="" k(number)\n");="" scanf("%d%d",&x,&k);="" for(i="0;i<n;i++)" {="" if(arr[i]="=x)" {="" p="i;" break;="" }="" }="" printf("\n%d="" closest="" elements="" to="" %d="" are="" ",k,x);="" for(i="0;i<k;i++)" {="" if(c="=k)" break;="" printf("="" %d",arr[p-(i+1)]);="" ++c;="" if(c="=k)" break;="" printf("="" %d",arr[p+(i+1)]);="" ++c;="" }="" }="">

• Reply • Share ›

Deepesh Panjabi • a year ago

http://ideone.com/sE3oXV

1 • Reply • Share ›

guest • a year ago

instead of taking n (length of the array) as parameter in printKClosest i tried to compute n inside the function and then use it but it does not work !! can anybody help me why do i have to take n as parameter nd i cannot compute it inside the function??

1 • Reply • Share ›

GOPI GOPINATH ➚ guest • a year ago

no we cannot do that.....we cannot find the size of an array inside a function...because inside a function sizeof(arr) will give the size of pointer instead of sizeof whole array

3 • Reply • Share ›

jimmy • a year ago

what if we take an array and store the difference between that element and the original array elements ..ans after that sort the array ...and start with 1 to k+1, add element to the diff. and print them....

1 • Reply • Share ›

GOPI GOPINATH • a year ago

2 • Reply • Share ›

Rajeev • a year ago

I think in example output should be 30, 39, 42,45

45 - 35 = 10 and 35 - 22 = 13 so 45 is more closer than 22

5 • Reply • Share ›

GeeksforGeeks Mod ➚ Rajeev • a year ago

Thanks for pointing this out. We have updated the example.

• Reply • Share ›

L_Earner ➚ GeeksforGeeks • a year ago

Right at top of discussion(where topic heading goes), You need to update the same as
39,30,42,45....it is misleading when you try to solve without going through solution given her

1 • Reply • Share ›

Aman Pandey • a year ago

Aman Pandey • a year ago

According to the logic in the above program :

```
while (l >= 0 && r < n && count < k)
{
if (x - arr[l] < arr[r] - x)
printf("%d ", arr[l--]);
else
printf("%d ", arr[r++]);
count++;
}
```

the output for : K = 4, X = 35

arr[] = {12, 16, 22, 30, 35, 39, 42,
45, 48, 50, 53, 55, 56}

should be : 39,30,42,45..

Isnt it? Because we are printing the value closest to 35.
Please correct me if i am wrong.

• Reply • Share ›

Ahmed Atito • a year ago

Hello,

I think there's something wrong in the first sample
while Output should be 39 30 42 45

since |35 - 45| < |35 - 22| therefore 45 is closer to 35 than 22

• Reply • Share ›

kv ↗ Ahmed Atito • a year ago

Distance is measured in terms of difference in positions rather than the difference between the numbers

• Reply • Share ›

Ahmed Atito ↗ kv • a year ago

If it's as you say
then why this calculation !!
simply binary search for X and return value to idx
print from idx - k / 2 to idx - k / 2 + k

• Reply • Share ›

Rajeev ↗ kv • a year ago

but solution that has been given is based on difference between numbers

• Reply • Share ›

dazerduzzy • a year ago

Hai
I have a small doubt.. Help me..
How about the K value ??
Is that always even value ?? if K = 3 then what is the output for X = 35 ??

Thanks..

• Reply • Share ›

Ahmed Atito dazerduzzy • a year ago

39 30 42

1 • Reply • Share ›

- 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

- lt_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