Given a sorted array and a number x, find the pair in array whose sum is closest to x

Given a sorted array and a number x, find a pair in array whose sum is closest to x.

Examples:

```
Input: arr[] = {10, 22, 28, 29, 30, 40}, x = 54
Output: 22 and 30

Input: arr[] = {1, 3, 4, 7, 10}, x = 15
Output: 4 and 10
```

A simple solution is to consider every pair and keep track of closest pair (absolute difference between pair sum and x is minimum). Finally print the closest pair. Time complexity of this solution is $O(n^2)$

An efficient solution can find the pair in O(n) time. The idea is similar to method 2 of this post. Following is detailed algorithm.

```
1) Initialize a variable diff as infinite (Diff is used to store the
   difference between pair and x).  We need to find the minimum diff.
2) Initialize two index variables l and r in the given sorted array.
       (a) Initialize first to the leftmost index:  l = 0
       (b) Initialize second  the rightmost index:  r = n-1
3) Loop while l < r.
       (a) If  abs(arr[l] + arr[r] - sum) < diff  then
           update diff and result
       (b) Else if(arr[l] + arr[r] <  sum )  then l++
       (c) Else r--
```

Following is C++ implementation of above algorithm.

C++

```
// Simple C++ program to find the pair with sum closest to a given no.
#include <iostream>
#include <climits>
#include <cstdlib>
using namespace std;
```

```
// Prints the pair with sum closest to x
void printClosest(int arr[], int n, int x)
{
    int res_l, res_r;  // To store indexes of result pair

    // Initialize left and right indexes and difference between
    // pair sum and x
    int l = 0, r = n-1, diff = INT_MAX;

    // While there are elements between l and r
    while (r > l)
    {
       // Check if this pair is closer than the closest pair so far
       if (abs(arr[l] + arr[r] - x) < diff)
       {
           res_l = l;
           res_r = r;
           diff = abs(arr[l] + arr[r] - x);
       }

       // If this pair has more sum, move to smaller values.
       if (arr[l] + arr[r] > x)
           r--;
       else // Move to larger values
           l++;
    }

    cout <<" The closest pair is " << arr[res_l] << " and " << arr[res_r];
}

// Driver program to test above functions
int main()
{
    int arr[] =  {10, 22, 28, 29, 30, 40}, x = 54;
    int n = sizeof(arr)/sizeof(arr[0]);
    printClosest(arr, n, x);
    return 0;
}
```

Run on IDE

Java

```
// Java program to find pair with sum closest to x
import java.io.*;
import java.util.*;
import java.lang.Math;

class CloseSum {

    // Prints the pair with sum cloest to x
    static void printClosest(int arr[], int n, int x)
    {
        int res_l=0, res_r=0;  // To store indexes of result pair

        // Initialize left and right indexes and difference between
        // pair sum and x
        int l = 0, r = n-1, diff = Integer.MAX_VALUE;

        // While there are elements between l and r
        while (r > l)
        {
            // Check if this pair is closer than the closest pair so far
            if (Math.abs(arr[l] + arr[r] - x) < diff)
            {
               res_l = l;
               res_r = r;
               diff = Math.abs(arr[l] + arr[r] - x);
            }

            // If this pair has more sum, move to smaller values.
```

```
            if (arr[l] + arr[r] > x)
                r--;
            else // Move to larger values
                l++;
        }

    System.out.println(" The closest pair is "+arr[res_l]+" and "+ arr[res_r]);
}


    // Driver program to test above function
    public static void main(String[] args)
    {
        int arr[] =  {10, 22, 28, 29, 30, 40}, x = 54;
        int n = arr.length;
        printClosest(arr, n, x);
    }
}
/*This code is contributed by Devesh Agrawal*/
```

Run on IDE

Output:

```
The closest pair is 22 and 30
```

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

GATE CS Corner Company Wise Coding Practice

Algorithms

(Login to Rate and Mark)

3 Average Difficulty : **3/5.0**
Based on **2** vote(s)

Add to TODO List

Mark as DONE

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

Share this post!