Smallest Difference pair of values between two unsorted Arrays

Difficulty Level : Easy • Last Updated : 18 May, 2021

Given two arrays of integers, compute the pair of values (one value in each array) with the smallest (non-negative) difference. Return the difference.

Examples :

```
Input : A[] = {1, 3, 15, 11, 2}
        B[] = {23, 127, 235, 19, 8}
Output : 3
That is, the pair (11, 8)

Input : A[] = {10, 5, 40}
        B[] = {50, 90, 80}
Output : 10
That is, the pair (40, 50)
```

by iterating through the arrays using the approach discussed in below post.
Find the closest pair from two sorted arrays

Consider the following two arrays:
A: {l, 2, 11, 15}
B: {4, 12, 19, 23, 127, 235}
1. Suppose a pointer a points to the beginning of A and a pointer b points to the beginning of B. The current difference between a and bis 3. Store this as the min.
2. How can we (potentially) make this difference smaller? Well, the value at bis bigger than the value at a, so moving b will only make the difference larger. Therefore, we want to move a.
3. Now a points to 2 and b (still) points to 4. This difference is 2, so we should update min. Move a, since it is smaller.
4. Now a points to 11 and b points to 4. Move b.
5. Now a points to 11 and b points to 12. Update min to 1. Move b. And so on.

Below is the implementation of the idea.

C++

```
// C++ Code to find Smallest
// Difference between two Arrays
#include <bits/stdc++.h>
```

```
{
    // Sort both arrays using
    // sort function
    sort(A, A + m);
    sort(B, B + n);

    int a = 0, b = 0;

    // Initialize result as max value
    int result = INT_MAX;

    // Scan Both Arrays upto
    // sizeof of the Arrays
    while (a < m && b < n)
    {
        if (abs(A[a] - B[b]) < result)
            result = abs(A[a] - B[b]);

        // Move Smaller Value
        if (A[a] < B[b])
            a++;

        else
            b++;
    }

    // return final sma result
    return result;
}
```

```
    // Input given array B
    int B[] = {4, 12, 19, 23, 127, 235};


    // Calculate size of Both arrays
    int m = sizeof(A) / sizeof(A[0]);
    int n = sizeof(B) / sizeof(B[0]);

    // Call function to print
    // smallest result
    cout << findSmallestDifference(A, B, m, n);

    return 0;
}
```

Java

```
// Java Code to find Smallest
// Difference between two Arrays
import java.util.*;

class GFG
{

    // function to calculate Small
    // result between two arrays
```

```
        Arrays.sort(B);

        int a = 0, b = 0;

        // Initialize result as max value
        int result = Integer.MAX_VALUE;

        // Scan Both Arrays upto
        // sizeof of the Arrays
        while (a < m && b < n)
        {
            if (Math.abs(A[a] - B[b]) < result)
                result = Math.abs(A[a] - B[b]);

            // Move Smaller Value
            if (A[a] < B[b])
                a++;

            else
                b++;
        }

        // return final sma result
        return result;
    }

    // Driver Code
    public static void main(String[] args)
    {
```

```
        // Calculate size of Both arrays
        int m = A.length;
        int n = B.length;

        // Call function to
        // print smallest result
        System.out.println(findSmallestDifference
                                (A, B, m, n));

    }
}
// This code is contributed
// by Arnav Kr. Mandal.
```

Python3

```
# Python 3 Code to find
# Smallest Difference between
# two Arrays
import sys

# function to calculate
# Small result between
# two arrays
def findSmallestDifference(A, B, m, n):
```

```
    b = 0

    # Initialize result as max value
    result = sys.maxsize

    # Scan Both Arrays upto
    # sizeof of the Arrays
    while (a < m and b < n):

        if (abs(A[a] - B[b]) < result):
            result = abs(A[a] - B[b])

        # Move Smaller Value
        if (A[a] < B[b]):
            a += 1

        else:
            b += 1
    # return final sma result
    return result

# Driver Code

# Input given array A
A = [1, 2, 11, 5]

# Input given array B
B = [4, 12, 19, 23, 127, 235]
```

```
print(findSmallestDifference(A, B, m, n))

# This code is contributed by
# Smitha Dinesh Semwal
```

C#

```
// C# Code to find Smallest
// Difference between two Arrays
using System;

class GFG
{

    // function to calculate Small
    // result between two arrays
    static int findSmallestDifference(int []A, int []B,
                                      int m, int n)
    {

        // Sort both arrays using
        // sort function
        Array.Sort(A);
        Array.Sort(B);

        int a = 0, b = 0;
```

```
        {
            if (Math.Abs(A[a] - B[b]) < result)
                result = Math.Abs(A[a] - B[b]);

            // Move Smaller Value
            if (A[a] < B[b])
                a++;

            else
                b++;
        }

        // return final sma result
        return result;
    }

    // Driver Code
    public static void Main()
    {

        // Input given array A
        int []A = {1, 2, 11, 5};

        // Input given array B
        int []B = {4, 12, 19, 23, 127, 235};


        // Calculate size of Both arrays
        int m = A.Length;
```

```
    }
}

// This code is contributed
// by nitin mittal.
```

PHP

```
<?php
// PHP Code to find Smallest
// Difference between two Arrays


// function to calculate Small
// result between two arrays
function findSmallestDifference($A, $B,
                                $m, $n)
{
    // Sort both arrays
    // using sort function
    sort($A);
    sort($A, $m);
    sort($B);
    sort($B, $n);

    $a = 0; $b = 0;
```

```
    // Scan Both Arrays upto
    // sizeof of the Arrays
    while ($a < $m && $b < $n)
    {
        if (abs($A[$a] - $B[$b]) < $result)
            $result = abs($A[$a] - $B[$b]);

        // Move Smaller Value
        if ($A[$a] < $B[$b])
            $a++;

        else
            $b++;
    }

    // return final sma result
    return $result;
}

// Driver Code
{
    // Input given array A
    $A = array(1, 2, 11, 5);

    // Input given array B
    $B = array(4, 12, 19, 23, 127, 235);


    // Calculate size of Both arrays
```

```
    return 0;
}

// This code is contributed by nitin mittal.
?>
```

Javascript

```
<script>

// JavaScript Code to find Smallest
// Difference between two Arrays

// function to calculate Small
// result between two arrays
function findSmallestDifference(A, B, m, n)
{

    // Sort both arrays using
    // sort function
    A.sort((a, b) => a - b);
    B.sort((a, b) => a - b);

    let a = 0, b = 0;

    // Initialize result as max value
```

```
        {
            if (Math.abs(A[a] - B[b]) < result)
                result = Math.abs(A[a] - B[b]);

            // Move Smaller Value
            if (A[a] < B[b])
                a++;
            else
                b++;
        }

        // Return final sma result
        return result;
    }

    // Driver Code

    // Input given array A
    let A = [ 1, 2, 11, 5 ];

    // Input given array B
    let B = [ 4, 12, 19, 23, 127, 235 ];

    // Calculate size of Both arrays
    let m = A.length;
    let n = B.length;

    // Call function to print
    // smallest result
    document.write(findSmallestDifference(
```

Output :

```
1
```

This algorithm takes O(m log m + n log n) time to sort and O(m + n) time to find the minimum difference. Therefore, the overall runtime is O(m log m + n log n).

This article is contributed by **Mr. Somesh Awasthi**. If you like GeeksforGeeks and would like to contribute, you can also write an article using write.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.
Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Next

Find minimum difference between any two elements

RECOMMENDED ARTICLES

Page :

Article Contributed By :

GeeksforGeeks

Vote for difficulty

Current difficulty : Easy

Article Tags : Arrays, Sorting

Practice Tags : Arrays, Sorting

Report Issue

GeeksforGeeks
5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org