**5 kyu**

**The maximum and minimum difference -- Challenge version**

112100% of 2162 of63[myjinxin2015](https://www.codewars.com/users/myjinxin2015)

JavaScript

* [TRAIN AGAIN](https://www.codewars.com/kata/the-maximum-and-minimum-difference-challenge-version/train/javascript)
* [NEXT KATA](https://www.codewars.com/trainer/javascript)

Details

[Solutions](https://www.codewars.com/kata/the-maximum-and-minimum-difference-challenge-version/solutions/javascript)

[Discourse (3)](https://www.codewars.com/kata/the-maximum-and-minimum-difference-challenge-version/discuss/javascript)

* Add to Collection
* |
* Share this kata:

When no more interesting kata can be resolved, I just choose to create the new kata, to solve their own, to enjoy the process --myjinxin2015 said

**Description:**

Given two array of integers(arr1,arr2). Your task is going to find a pair of numbers(an element in arr1, and another element in arr2), their difference is as big as possible(absolute value); Again, you should to find a pair of numbers, their difference is as small as possible. Return the maximum and minimum difference values by an array: [ max difference, min difference ]

For example:

Given arr1 = [3,10,5], arr2 = [20,7,15,8]

should return [17,2] because 20 - 3 = 17, 10 - 8 = 2

**Note:**

* arr1 and arr2 contains only integers(positive, negative or 0);
* arr1 and arr2 may have different lengths, they always has at least one element;
* All inputs are valid.
* This is a challenge version, Please optimize your algorithm to avoid time out ;-)
* If you feel difficult, please [try the simple version](https://www.codewars.com/kata/583c5469977933319f000403).

**About testcases**

* Basic test: 5 testcases
* Random test1: 100 testcases, arr1 and arr2 contains 1-20 elements
* Random test2: 300 testcases, arr1 and arr2 contains 10000 elements

**Some Examples**

maxAndMin([3,10,5],[20,7,15,8]) === [17,2]

maxAndMin([3],[20]) === [17,17]

maxAndMin([3,10,5],[3,10,5]) === [7,0]

maxAndMin([1,2,3,4,5],[6,7,8,9,10]) === [9,1]

<https://www.codewars.com/kata/the-maximum-and-minimum-difference-challenge-version/javascript>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Numerics;

namespace ConsoleApp1

{

class Program

{

static int[] max\_and\_min(int[] arr1, int[] arr2)

{

/\*

Since the arrays are sorted, you can pass through them with 2

pointers(one for each array).

If | A[i + 1] - B[j] | < | A[i] - B[j + 1] | then increment i,

otherwise increment j.

Continue until you've reached the end of one of the arrays.

Keep track of the minimal indexes as you go.

\*/

Array.Sort(arr1);

Array.Sort(arr2);

//int max\_difference = Math.Max(max1 - min2, max2 - min1);

int max\_difference = Math.Max(arr1[arr1.Length-1] - arr2[0], arr2[arr2.Length-1] - arr1[0]);

int i = 0;

int j = 0;

//int dif = int.MaxValue;

int min\_difference = Math.Abs(arr1[0] - arr2[0]);

int diff1 = min\_difference;

int diff2 = min\_difference;

while (i < arr1.Length && j < arr2.Length)

{

if (i + 1 < arr1.Length) diff1 = Math.Abs(arr1[i + 1] - arr2[j]);

if (j + 1 < arr2.Length) diff2 = Math.Abs(arr1[i] - arr2[j + 1]);

if (diff1 < diff2) i++;

else j++;

min\_difference = Math.Min(min\_difference, diff1);

min\_difference = Math.Min(min\_difference, diff2);

}

return new int[] { max\_difference, min\_difference };

// [max difference, min difference]

}

static void Main(string[] args)

{

int[] arr1 = { 3, 10, 5 };

int[] arr2 = { 20, 7, 15, 8 };

//int[] input = { 1, 5, 3, 19, 18, 25 };

//max\_and\_min( )

//1852 14

//int[] res = max\_and\_min(new int[] { -870, 91, -141, -739, 707, -803, -195, -963, 99, 861 }, new int[]{796, -468, 889, 58, -765, -901, -311, -399, -764, -181, 841, -670, -589});

//[7,0]

//int[] res = max\_and\_min(new int[] { 3, 10, 5,90 },new int[]{ 7,8,9, 90});

//int[] arr1 = { -51954, 89153, -83318, 86139, 95438, -93409, 61210, 40173, -79657, -82367, -59485, 47584, 95311, -45142, 23373, -49057, -91951, -85803, -95827, -6147, 69988, -65471, 47217, 18938, 12833, -9887, -47770, 31108, 31780, -37910, 1320, 23733, 91498, -68843, -61851, -89160, -12768, 81745, 61100, -70902, -92788, -53758, 83233 };

//int[] arr2 = {25573, 12157, -89381, 95800, -39003, 18029, -10608, 39604, 50764, 23094, -3574, 60093, 26125};

//Expected: [191627, 221], instead got: [191627, 1093]

int[] res = max\_and\_min(arr1, arr2);

Console.WriteLine(res[0] + " " + res[1]);

Console.ReadLine();

}

}

}

<script>

**function** maxAndMin(arr1,arr2)

{

     arr1.sort(**function**(a, b){**return** a - b});

     arr2.sort(**function**(a, b){**return** a - b});

**var** max\_difference = Math.max(arr1[arr1.length-1] - arr2[0], arr2[arr2.length-1] - arr1[0]);

**var** i = 0;

**var** j = 0;

**var** min\_difference = Math.abs(arr1[0] - arr2[0]);

**var** diff1 = min\_difference;

**var** diff2 = min\_difference;

     while (i < arr1.length && j < arr2.length)

     {

**if** (i + 1 < arr1.length) diff1 = Math.abs(arr1[i + 1] - arr2[j]);

**if** (j + 1 < arr2.length) diff2 = Math.abs(arr1[i] - arr2[j + 1]);

**if** (diff1 < diff2) i++;

**else** j++;

          min\_difference = Math.min(min\_difference, diff1);

          min\_difference = Math.min(min\_difference, diff2);

     }

**return** [ max\_difference, min\_difference ];

}

console.log( maxAndMin([3,10,5],[20,7,15,8]));

</script>