**6 kyu**

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

10393% of 7968 of256[myjinxin2015](https://www.codewars.com/users/myjinxin2015)

Python

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

Details

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

[Discourse (12)](https://www.codewars.com/kata/the-maximum-and-minimum-difference-simple-version/discuss/python)

* 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 simple version, if you want some challenges, [try the challenge version](https://www.codewars.com/kata/583c592928a0c0449d000099).

**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-simple-version/python>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Numerics;

namespace ConsoleApp1

{

class Program

{

private static void quicksort(int[] vector, int[] paralelo, int primero, int ultimo)

{

int i, j, central;

double pivote;

central = (primero + ultimo) / 2;

pivote = vector[central];

i = primero;

j = ultimo;

do

{

while (vector[i] < pivote) i++;

while (vector[j] > pivote) j--;

if (i <= j)

{

int temp;

temp = vector[i];

vector[i] = vector[j];

vector[j] = temp;

int temp2 = paralelo[i];

paralelo[i] = paralelo[j];

paralelo[j] = temp2;

i++;

j--;

}

} while (i <= j);

if (primero < j)

{

quicksort(vector, paralelo, primero, j);

}

if (i < ultimo)

{

quicksort(vector, paralelo, i, ultimo);

}

}

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

{

//pass

int i = 0;

int min1 = arr1[0], max1 = arr1[0];

for ( i = 1; i < arr1.Length; i++)

{

if (arr1[i] > max1)

{

max1 = arr1[i];

}

if (arr1[i] < min1)

{

min1 = arr1[i];

}

}

int min2 = arr2[0], max2 = arr2[0];

for ( i = 1; i < arr2.Length; i++)

{

if (arr2[i] > max2)

{

max2 = arr2[i];

}

if (arr2[i] < min2)

{

min2 = arr2[i];

}

}

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

//Console.WriteLine(max\_difference);

int[] a = new int[arr1.Length + arr2.Length];

int[] marcas = new int[a.Length];

for( i =0; i< arr1.Length; i++)

{

a[i] = arr1[i];

marcas[i] = 1;

}

for( i =0; i<arr2.Length; i++)

{

a[i + arr1.Length] = arr2[i];

marcas[i + arr1.Length] = 2;

}

//for(int i =0; i<a.Length; i++)

//{

// Console.Write(a[i] + " ");

//}

quicksort(a, marcas, 0, a.Length - 1);

//for( i =0; i<a.Length; i++)

//{

// Console.WriteLine(a[i] + " " + marcas[i]);

//}

int min\_difference = int.MaxValue;

i = 0;

while (i< a.Length)

{

int j = i + 1;

while(j < a.Length && marcas[i] == marcas[j])

{

j++;

}

if (j < a.Length && marcas[i] != marcas[j])

{

min\_difference = Math.Min(min\_difference, Math.Abs(a[i] - a[j]));

}

i++;

}

//Console.WriteLine(min\_difference);

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

// [max difference, min difference]

}

static void Main(string[] args)

{

//max\_and\_min(new int[] { 3, 10, 5 }, new int[] { 20, 7, 15, 8 });

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

//max\_and\_min( )

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});

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

Console.ReadLine();

}

}

}

**def** quick\_sort(vector, paralelo, primero, ultimo):

    i =0

    j = 0

    central =0

    pivote = 0.0

    central = int((primero + ultimo)/2 )

    pivote = vector[central]

    i = primero

    j = ultimo

**while**(True):

**while**(vector[i] < pivote): i+=1

**while**(vector[j] > pivote): j-=1

**if**(i<=j):

            temp = vector[i]

            vector[i] = vector[j]

            vector[j] = temp

            temp2 = paralelo[i]

            paralelo[i] = paralelo[j]

            paralelo[j] =temp2

            i += 1

            j -= 1

**if**(i > j):

**break**

**if**(primero < j): quick\_sort(vector, paralelo, primero, j)

**if**(i < ultimo): quick\_sort(vector, paralelo, i, ultimo)

**def** max\_and\_min(arr1,arr2):

    min1 = arr1[0]

    max1 = arr1[0]

**for** i **in** range(1, len(arr1)):

**if**(arr1[i] > max1): max1 = arr1[i]

**if**(arr1[i] < min1): min1 = arr1[i]

    min2 = arr2[0]

    max2 = arr2[0]

**for** i **in** range(1, len(arr2)):

**if**(arr2[i] > max2): max2 = arr2[i]

**if**(arr2[i] < min2): min2 = arr2[i]

    max\_difference = max(max1 - min2, max2 - min1)

    a = [0] \* (len(arr1) + len(arr2))

    marcas = [1 ] \* len(a)

**for** i **in** range(0, len(arr1)):

        a[i] = arr1[i]

        marcas[i] = 1

**for** i **in** range(0, len(arr2)):

        a[i+len(arr1)]  = arr2[i]

        marcas[i + len(arr1)] = 2

    quick\_sort(a, marcas, 0, len(a) - 1)

    min\_difference = 10000000000000000

    i = 0

**while**(i <len(a)):

        j = i+1

**while**(j < len(a) **and** marcas[i] == marcas[j]): j+=1

**if**(j < len(a) **and** marcas[i] != marcas[j]) :

            min\_difference = min(min\_difference, abs(a[i] - a[j]));

        i+=1

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

ar1 = [-870,91,-141,-739,707,-803,-195,-963,99,861]

ar2 = [796,-468,889,58,-765,-901,-311,-399,-764,-181,841,-670,-589]

**print**(max\_and\_min(ar1,ar2)) *# [1852, 14])*