

Forum:[Hack Forums](#)

Thread: [Creating a Method to find Max Difference in an Array.](#)

SIM-810273 - November 18, 2015, 11:31 am

Hi all, I have to create a function that will look at an array of 5 numbers and find the difference between adjacent numbers and from them differences then return the largest difference.

Honestly I'm not sure where to start I know using loops will be my best bet but I'm unsure.

Arrays

```
Code int [] numbers = {12, 8, 34, 10, 59};
```

```
int [] numbers2 = {-50, 100, 20, -40};
```

So for the first one the maximum difference would be 49 (10 to 59)

Second one the maximum difference would be 150 (-50 to 100)

Remember it has to be between numbers next to each other.

Thanks in advance!

Sparx710 - November 18, 2015, 11:51 am

If the array always contains 5 numbers, then you can hard code the comparison of the numbers.

Calculate `Math.abs(numbers[0]-numbers[1])`, `Math.abs(numbers[1]-numbers[2])` and so on.

Then print/return the largest one.

Or loop through the array until `numbers.length-1` and calculate `Math.abs(numbers[i]-numbers[i+1])`

As I cannot write PMs yet, here is my response to your message^^

You can compare a value in an array to the next one like i wrote above (`array[index]` and `array[index+1]`)

PHP Code

```
<?php
public static int maxDifference(int[] a) {
    int maxDiff = 0;
    //Loop until the second last value, so you can compare it with the last one at the end
    for (int index = 0; index < a.length - 1; index++) {
        int diff = a[index] - a[index + 1];
    //Calculate the difference of two numbers, that are next to each other
        if (diff < 0) diff = diff * -1; //Get a postive value
        if (diff > maxDiff) maxDiff = diff; //Set new maxDiff
    }
    return maxDiff;
}
```

WainWright - November 19, 2015, 10:48 am

I can imagine many ways to do this but the simplest one I have is to simply sort the array in ascending order and then subtract the last element from the first.

```
Code Arrays.sort(numbers); //Arrays.parallelSort(numbers);
```

```
return numbers[4] - numbers[0]; //return numbers[numbers.length - 1] - numbers[0];
```

I don't know if performance would be better to manually find the largest and smallest or not. Sometimes the methods Oracle provides are more efficient.

FocalBanger - November 20, 2015, 6:35 am

Iterate through the array and keep track of the largest number (ex. if(biggestNumber < array[i]){biggestNumber = array[i]}) then keep track of the biggest difference by subtracting the biggest number from each element in the array.

Enchanter - November 20, 2015, 10:51 am

In Java8 you can do

```
Code public static int getMaxAd(int[] arr){  
    return Arrays.stream(IntStream.range(0, arr.length - 1).toArray())  
        .map(x -> arr[x + 1] - arr[x])  
        .max().getAsInt();  
}
```

So

```
Code int max = getMaxAd(numbers);
```