**5 kyu**

**Maximum subarray sum**

70514888% of1,4152,226 of11,789[knotman90](https://www.codewars.com/users/knotman90)

Java

* [TRAIN AGAIN](https://www.codewars.com/kata/maximum-subarray-sum/train/java)
* [NEXT KATA](https://www.codewars.com/trainer/java)

Details

[Solutions](https://www.codewars.com/kata/maximum-subarray-sum/solutions/java)

[Forks (10)](https://www.codewars.com/kata/maximum-subarray-sum/forks/java)

[Discourse (163)](https://www.codewars.com/kata/maximum-subarray-sum/discuss/java)

* Add to Collection
* |
* Share this kata:

The maximum sum subarray problem consists in finding the maximum sum of a contiguous subsequence in an array or list of integers:

Max.sequence(new int[]{-2, 1, -3, 4, -1, 2, 1, -5, 4});

// should be 6: {4, -1, 2, 1}

Easy case is when the list is made up of only positive numbers and the maximum sum is the sum of the whole array. If the list is made up of only negative numbers, return 0 instead.

Empty list is considered to have zero greatest sum. Note that the empty list or array is also a valid sublist/subarray.

<https://www.codewars.com/kata/maximum-subarray-sum/java>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

public static int sequence(int[] arr)

{

//return 4;

int max\_sum = arr[0];

int curr\_sum = arr[0];

for (int i = 1; i < arr.Length; i++)

{

curr\_sum = Math.Max(arr[i], curr\_sum + arr[i]);

max\_sum = Math.Max(max\_sum, curr\_sum);

}

return max\_sum;

}

static void Main(string[] args)

{

Console.WriteLine(sequence(new int[] { -2, 1, -3, 4, -1, 2, 1, -5, 4 }));

Console.ReadLine();

}

}

}