You have an array of integers, and each integer denotes the length of a segment. Determine whether it is possible to pick three segments and construct a triangle out of them.

*It is possible to construct a triangle if the combined length of the two shortest segments is greater than the length of the longest segment.*

**Example**

* For segmentsLength = [1, 2, 3, 4], the output should be  
  triangularity(segmentsLength) = true.

The segments of lengths 2, 3, and 4 can be used to form a triangle.

* For segmentsLength = [1, 2, 3], the output should be  
  triangularity(segmentsLength) = false.

The given three segments don't form a triangle.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] array.integer segmentsLength**

An array of segment lengths.

*Guaranteed constraints:*  
3 ≤ segmentsLength.length ≤ 105,  
1 ≤ segmentsLength[i] ≤ 2 · 109.

* **[output] boolean**

Return true if you can construct a triangle using three segments from segmentsLength, otherwise return false.

<https://codefights.com/challenge/xunq4T96b3fY54Wif?utm_source=emailNotification&utm_medium=email&utm_campaign=featuredChallenge>

static bool triangularity(int[] segmentsLength)

{

Array.Sort(segmentsLength);

for (int i = 0; i < segmentsLength.Length - 2; i++)

{

if (segmentsLength[i] + segmentsLength[i + 1] > segmentsLength[i + 2])

{

return true;

}

}

return false;

}