Given a sequence, check whether it is [cyclic increasing](keyword://cyclic-increasing-sequence).

**Example**

* For sequence = [5, 9, 1, 2, 4], the output should be  
  cyclicSequence(sequence) = true.

Increasing sequence can be started from number 1.

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

**Input/Output**

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

*Guaranteed constraints:*  
1 ≤ sequence.length ≤ 105,  
-105 ≤ sequence[i] ≤ 105.

* **[output] boolean**

true if sequence is cyclic increasing, falseotherwise.

**[C#] Syntax Tips**

// Prints help message to the console

// Returns a string

string helloWorld(string name) {

Console.Write("This prints to the console when you Run Tests");

return "Hello, " + name;

}

<https://codefights.com/challenge/jvjiAst2kvxyxzJ3j/solutions>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static bool cyclicSequence(int[] sequence)

{

int ind\_min = Array.IndexOf(sequence, sequence.Min());

//int[] sort = new int[sequence.Length];

List<int> sort = new List<int>();

for (int i = ind\_min; i < sequence.Length; i++)

{

sort.Add(sequence[i]);

}

for (int i = 0; i < ind\_min; i++)

{

sort.Add(sequence[i]);

}

//foreach (int elem in sort)

//{

// Console.Write(elem + " ");

//}

for (int i = 0; i < sort.Count - 1; i++)

{

if (sort[i] >= sort[i + 1]) return false;

}

return true;

}

static void Main(string[] args)

{

// int[] sequence = { 5, 9, 1, 2, 4 };

//int[] sequence = { 1, 3, 2 };

// int[] sequence = { 5, -4, -3, 0, 1, 3, 4, 5 };

int[] sequence = { 5, 9, 1, 2, 4 };

Console.WriteLine( cyclicSequence(sequence));

//Console.WriteLine(cyclicSequence(sequence));

Console.ReadLine();

}

}

}

static bool cyclicSequence(int[] sequence)

{

int ind\_min = Array.IndexOf(sequence, sequence.Min());

int[] sort = new int[sequence.Length];

Array.Copy(sequence, ind\_min, sort, 0, sequence.Length - (ind\_min));

Array.Copy(sequence, 0, sort, sequence.Length - (ind\_min), ind\_min);

for (int i = 0; i < sort.Length - 1; i++) if (sort[i] >= sort[i + 1]) return false;

return true;

}