**6 kyu**

**Simple Fun #208: Find Sub Array With Same Element**

2097% of 1934[myjinxin2015](https://www.codewars.com/users/myjinxin2015)

JavaScript

* [TRAIN AGAIN](https://www.codewars.com/kata/simple-fun-number-208-find-sub-array-with-same-element/train/javascript)
* [NEXT KATA](https://www.codewars.com/trainer/javascript)

Details

[Solutions](https://www.codewars.com/kata/simple-fun-number-208-find-sub-array-with-same-element/solutions/javascript)

[Discourse](https://www.codewars.com/kata/simple-fun-number-208-find-sub-array-with-same-element/discuss/javascript)

* Add to Collection
* |
* Share this kata:

**Task**

Given an array of positive integers a and an integer k, find the first and last index of the longest subarray of a that consists only of k.

If the array contains multiple subarrays of the same length, return indices of the last one.

If k doesn't exist in a, return [-1,-1].

**Input/Output**

* [input] integer array a

A non-empty array. For each valid i 1 ≤ a[i] ≤ 9.

* [input] integer k

1 ≤ k ≤ 9.

* [output] an integer array

The first and the last indices of the longest subarray consisting of k only, as an array in the format [start,end], or [-1, -1]if k is not present in a.

**Example**

For a = [2,1,1,1,1,3,3,4,5,1,1,1] and k = 3,

the output should be [5,6].

The longest subarray of a that contains 3s starts at index 5 and ends at index 6.

For a = [2,1,1,1,1,3,3,4,5,1,1,1,1] and k = 1,

the output should be [9, 12].

There are two subarrays of 1, and they have equal length. One goes from index 1 to 4, and another one from index 9 to 12. The answer should be [9,12] as it is last to occur.

For a = [1, 2, 3] and k = 9,

the output should be [-1, -1].

<https://www.codewars.com/kata/simple-fun-number-208-find-sub-array-with-same-element/javascript>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

static int[] findSubArrayWithSameElement(int[] a, int k)

{

int start = -1, end = -1;

int max\_start = -1, max\_end = -1;

int max\_cont = 0;

//for (int i = a.Length - 1; i >= 0; i--)

//int pre = a[0];

bool flag = true;

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

{

if (a[i] == k)

{

if (flag)

{

start = i;

flag = false;

}

}

if (flag == false && start > -1 && k != a[i])

{

end = i - 1;

if (end > start)

{

if (end - start + 1 >= max\_cont)

{

max\_cont = end - start + 1;

max\_start = start;

max\_end = end;

}

}

flag = true;

}

}

if (start > end)

{

//max\_cont = end - start + 1;

end = a.Length - 1;

if (end - start + 1 >= max\_cont)

{

max\_start = start;

max\_end = end;

}

}

if(start == end)

{

if (start - end + 1 > max\_cont)

{

max\_start = start;

max\_end = end;

}

}

return new int[] { max\_start, max\_end };

}

// Driver code

public static void Main()

{

// Test.assertDeepEquals(

// findSubArrayWithSameElement(

//int[] a = { 2, 1, 1, 1, 1, 3, 3, 4, 5, 1, 1, 1, 1};

//int k = 1;

//,[9, 12])

//int[] a = { 1, 1, 1 };

//int k = 1;

int[] a = { 2, 4, 7, 1, 9, 8 };

int k = 4;

//int[] a = { 7, 7, 7, 1, 3, 4, 5, 5, 5, 5, 4, 3, 6, 9, 4, 6, 2, 9, 4, 8, 4, 7, 7, 8, 2, 9, 5, 6, 4, 9, 5, 9, 1, 6, 2, 1, 3, 8, 5, 2, 8, 6, 1, 1, 3, 7, 8, 8, 3, 2, 7, 6, 9, 4, 9, 7, 7, 5, 7, 8, 9, 1, 5, 2, 2, 1, 1, 4, 2, 1, 6, 9, 2, 4, 3, 3, 2, 6, 4 };

//int k = 7;

int[] res = findSubArrayWithSameElement(a, k);

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

Console.ReadLine();

}

}

}

<script>

function findSubArrayWithSameElement(a, k) {

//coding and coding..

var start = -1;

var end = -1;

var max\_start = -1;

var max\_end = -1;

var max\_cont = 0;

var flag = true;

for (let i = 0; i < a.length; i++)

{

if (a[i] == k)

{

if (flag)

{

start = i;

flag = false;

}

}

if (flag == false && start > -1 && k != a[i])

{

end = i - 1;

if (end > start)

{

if (end - start + 1 >= max\_cont)

{

max\_cont = end - start + 1;

max\_start = start;

max\_end = end;

}

}

flag = true;

}

}

if (start > end)

{

end = a.length - 1;

if (end - start + 1 >= max\_cont)

{

max\_start = start;

max\_end = end;

}

}

return [ max\_start, max\_end ];

}

document.write(findSubArrayWithSameElement( [2, 1, 1, 1, 1, 3, 3, 4, 5, 1, 1, 1], 3 ));

</script>