You're given an array of unsorted integers arr. Find the maximum length of a [contiguous subarray](keyword://contiguous-subarray) that contains at most 2 different integers (possibly several times).

**Example**

For arr = [7, 4, 5, 4, 4, 6], the output should be  
findBiSlice(arr) = 4.

The longest *contiguous subarray* that consists of only2 integers is [4, 5, 4, 4], and it contains 4elements.

**Input/Output**

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

Array of integers.

*Constraints:*  
2 ≤ arr.length ≤ 8000,  
1 ≤ arr[i] ≤ 104.

* **[output] integer**

The length of the maximal *contiguous subarray*containing no more than 2 integers.

<https://codefights.com/challenge/n75eG9MuhDzejuCyp/main?utm_source=challengeOfTheDay&utm_medium=email&utm_campaign=email_notification>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static int findBiSlice(int[] arr)

{

int max\_len = 0;

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

{

int elems = 0;

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

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

{

if (!total.Contains(arr[j]))

{

total.Add(arr[j]);

}

if (total.Count > 2)

{

break;

}

elems++;

}

max\_len = Math.Max(max\_len, elems);

}

return max\_len;

}

static void Main(string[] args)

{

int[] arr = { 7, 4, 5, 4, 4, 6 };

Console.WriteLine(findBiSlice(arr));

Console.ReadLine();

}

}

}