You just heard about a new Pokemon tournament. In order to compete in the tournament, you must follow these rules:

* You must use 2 Pokemon;
* Their combined attack power must be equal to maxPower.

You and your friends would like to compete, but there is limited time to enter! You need to pick the first 2 Pokemon in your list whose combined attack power is equal to maxPower and return their indices (0-based).  
If there are no such pairs, return an empty array.

Some of your friends are true Pokemon masters, and have hundreds of thousands of Pokemon. Make sure to help them find their Pokemon fast too!

**Example**  
For pokemonList = [4, 3, 2, 3, 4] and maxPower = 6,  
the output should be  
teamFinder(pokemonList, maxPower) = [0, 2].  
The pairs that add up to maxPower are:  
(4, 2), (3, 3), (2, 4).  
The pair that gets completed first is (4, 2) whose indices are [0, 2].

**Input/Output**

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

A list of the attack powers of your Pokemon.

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

* **[input] integer maxPower**

The combined attack powers of the 2Pokemon you choose must add up to maxPower.

*Guaranteed constraints:*  
2 ≤ maxPower ≤ 2 · 105.

* **[output] array.integer**

The indices of the pair of Pokemon whose combined attack power is equal to maxPowerand appear earliest in the list.

<https://codefights.com/challenge/af2y4DBXqibqmHyAN/solutions>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static int[] teamFinder(int[] pokemonList, int maxPower)

{

int[] indices = new int[pokemonList.Length];

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

{

indices[i] = i;

}

int[] copia = pokemonList.ToArray();

Array.Sort(copia, indices);

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

//{

// Console.Write(copia[i] + " ");

//}

//Console.WriteLine();

//Console.WriteLine("indices:");

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

//{

// Console.Write (indices[i] + " ");

//}

//Console.WriteLine();

List<int[]> lista = new List<int[]>();

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

{

int a = copia[i];

int indice\_b= Array.BinarySearch(copia, maxPower - a);

if (indice\_b >=0 && indice\_b != i )

{

lista.Add(new int[]{ Math.Min(indices[i], indices[ indice\_b]), Math.Max(indices[ i], indices[ indice\_b])});

}

}

Dictionary<int, int> diccio = new Dictionary<int, int>();

foreach (int[] elem in lista)

{

diccio[elem[0]] = elem[1];

}

foreach (var item in diccio.OrderBy(i => i.Value))

{

//Console.WriteLine(item);

return new int[] { item.Key, item.Value };

}

//foreach (int[] elem in lista)

//{

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

//}

return new int[] { };

}

static void Main(string[] args)

{

//int[] pokemonList= {4, 3, 2, 3, 4};

//int maxPower = 6;

//int[] pokemonList= {33, 17, 21, 31, 15, 27, 29, 35, 23, 25, 19};

//int maxPower = 27;

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

//int maxPower = 2;

int[] pokemonList= {10, 5, 2, 3, 7, 5};

int maxPower = 10;//[3, 4]

int[] res = teamFinder(pokemonList, maxPower);

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

Console.ReadLine();

}

}

}

Otras soluciones:

int[] teamFinder(int[] p, int m)

{

var z = new Dictionary<int, int>();

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

{

int t = p[i];

if (z.ContainsKey(m - t))

return new int[] { z[m - t], i };

if (!z.ContainsKey(t))

z[t] = i;

}

return new int[] { };

}