



Desafio: Análise Big O

```
// T: O(1), S: O(1)
int SumRandom(int[] a, int[] b)
{
    var rnd = new Random();
    return a[rnd.Next(0, a.Length)] + b[rnd.Next(0, b.Length)];
}
```

```
// T: O(N), S: O(1)
void MultiplyArray(int[] arr)
{
    for (int i = 0; i < arr.Length; i++)
    {
        arr[i] *= arr[i];
    }
}
```

```
// T: O (log N), S: O(1)
void SplitSum(int n)
{
    int a = 0, i = n;
    while (i > 0)
    {
        a += i;
        i /= 2;
    }
}
```

```
// T: O (n^2), S: O(1)
int SumNested(int n)
{
    int a = 0;
    var i = 0;
    while (i < n)
    {
        ++i;
        for (var j = n; j > i; j--)
        {
            a = a + i + j;
        }
    }
    return a;
}
```

```
// T: O (n + m), S: O(1)
int RandomSumTwoArrays(int n, int m)
{
    int a = 0, b = 0;
    var rnd = new Random();
    for (var i = 0; i < n; i++)
    {
        a = a + rnd.Next();
    }
    for (var j = 0; j < m; j++)
    {
        b = b + rnd.Next();
    }
    return a + b;
}
```

```

// T: O (log N), S: O(1)
int FindElementIndexInSortedArray(int[] arr, int searchElement)
{
    //assume-se que arr é ordenado
    var start = 0;
    var end = arr.Length - 1;
    var i = (int)((start + end) * 0.5f);

    while (end - start > 1)
    {
        var element = arr[i];
        if (element == searchElement)
        {
            return i;
        }
        else if (searchElement > element)
        {
            start = i;
        }
        else
        {
            end = i;
        }

        i = (int)((start + end) * 0.5f);
    }

    return arr[start] == searchElement ? start :
        arr[end] == searchElement ? end : -1;
}

```

```

// T: O(n), S: O(1)
int FindFirstIndexOfNumber(int number, int[] array)
{
    for (var i = 0; i < array.Length; i++)
    {
        if (array[i] == number)
        {
            return i;
        }
    }
    return -1;
}

```

```
// T: O(n), S: O(n)
int[] FindEachIndexOfNumber(int number, int[] array)
{
    var arrayOfIndexes = new List<int>(array.Length);
    for (int i = 0; i < array.Length; i++)
    {
        var element = array[i];
        if (element == number)
        {
            arrayOfIndexes.Add(i);
        }
    }
    return arrayOfIndexes.ToArray();
}
```

```
// T: O(n*m), S: O(n)
int[] ReturnDups(int[] array1, int[] array2)
{
    var size = System.Math.Max(array.Length, array2.Length);
    var dupeArray = new List<int>(size);
    foreach (var element in array1)
    {
        if (array2.Contains(element))
        {
            dupeArray.Add(element);
        }
    }
    return dupeArray.ToArray();
}
```

```
// T: O(n^2), S: O(1)
void SortByValue(int[] array)
{
    void Swap(int[] array, int index1, int index2)
    {
        var temp = array[index1];
        array[index1] = array[index2];
        array[index2] = temp;
    }

    var count = 1;
    while (count < array.Length)
    {
        var swapCount = 0;
        for (var i = 0; i < array.Length - count; i++)
        {
            if (array[i] > array[i + 1])
            {
                Swap(array, i, i + 1);
                swapCount++;
            }
        }
        count++;
    }
}
```

