1 . **What is the expected running time of the following C# code? Explain why. Assume the array's size is n.**

**long Compute(int[] arr)**

**{**

**long count = 0;**

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

**{**

**int start = 0, end = arr.Length-1;**

**while (start < end)**

**if (arr[start] < arr[end])**

**{ start++; count++; }**

**else**

**end--;**

**}**

**return count;**

**}**

Външният for цикъл ще се изпълни n пъти – размера на масива. В while цикъла на всяка стъпка или увеличаваме start или намаляваме end, но и в двата случая разликата между двете променливи намалява с 1. Тоест while ще се изпълни n – 1 пъти. Сложността еО(n \* (n – 1)) = О(n^2 – n) = О(n^2). В случая –n почти не е от значение и може да се пропусне.

1. **What is the expected running time of the following C# code? Explain why.**

**long CalcCount(int[,] matrix)**

**{**

**long count = 0;**

**for (int row=0; row<matrix.GetLength(0); row++)**

**if (matrix[row, 0] % 2 == 0)**

**for (int col=0; col<matrix.GetLength(1); col++)**

**if (matrix[row,col] > 0)**

**count++;**

**return count;**

**}**

**Assume the input matrix has size of n \* m.**

Външният for се изпълнява n на брой пъти. Тъй като всеки път, if условието може да бъде изпълнено в най-лошия случай и за него for цикъла се изпълнява m пъти. Следователно имаме сложност O(n\* m).

1. **\* What is the expected running time of the following C# code? Explain why.**

**long CalcSum(int[,] matrix, int row)**

**{**

**long sum = 0;**

**for (int col = 0; col < matrix.GetLength(0); col++)**

**sum += matrix[row, col];**

**if (row + 1 < matrix.GetLength(1))**

**sum += CalcSum(matrix, row + 1);**

**return sum;**

**}**

**Console.WriteLine(CalcSum(matrix, 0));**

**Assume the input matrix has size of n \* m.**

За тази задача трябва да се вметне, че използванията на **matrix.GetLength()** са объркани.Ако поправим решението, то for цикъла се изпълнява m (броя на колоните) пъти. Рекурсията започва от 0 и достига n – 1 ред - изпълнява се n пъти. Следователно сложността е О(n \* m).