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

1. **long Compute(int[] arr)**
2. **{**
3. **long count = 0;**
4. **for (int i=0; i<arr.Length; i++)**
5. **{**
6. **int start = 0, end = arr.Length-1;**
7. **while (start < end)**
8. **if (arr[start] < arr[end])**
9. **{ start++; count++; }**
10. **else**
11. **end--;**
12. **}**
13. **return count;**
14. **}**

**Отговор:** Външният цикъл се изпълнява n пъти, като на всяка итерация изпълняваме вътрешния по n пъти (всеки път разликата end - start намалява с 1, докато не стане 0) => O(n\*n)

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

1. **long CalcCount(int[,] matrix)**
2. **{**
3. **long count = 0;**
4. **for (int row=0; row<matrix.GetLength(0); row++)**
5. **if (matrix[row, 0] % 2 == 0)**
6. **for (int col=0; col<matrix.GetLength(1); col++)**
7. **if (matrix[row,col] > 0)**
8. **count++;**
9. **return count;**
10. **}**

**Отговор:** O(n-z + z\*m), където z е броя четни числа в първата колона на matrix. **Причина:** външният цикъл се изпълнява n пъти, а във вътрешния се влиза точно когато първото числото в реда е четно => z пъти. Ако z се приеме за константа, имаме О(n + m); ако z е от порядъка на n => O(n\*m)

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

1. **long CalcSum(int[,] matrix, int row)**
2. **{**
3. **long sum = 0;**
4. **for (int col = 0; col < matrix.GetLength(0); col++)**
5. **sum += matrix[row, col];**
6. **if (row + 1 < matrix.GetLength(1))**
7. **sum += CalcSum(matrix, row + 1);**
8. **return sum;**
9. **}**
10. **Console.WriteLine(CalcSum(matrix, 0));**

Ако броят редове n е по-голям от броя колони m, това ще се случи в първия цикъл; иначе програмата ще го изпълни n пъти, след което ще рекурсира - общият брой рекурсивни извиквания ще е n (докато row не излезе извън матрицата) => O(n^2).