Data Structures and Algorithms and Complexity Homework

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 цикъл ще се изпълни n пъти. Вътрешния while ще се изпълни от start до end пъти, което отново е n пъти (дължината на масива). Всеки път се случва едно от двете – или се увеличава start или се намаля end – и в двата случея двете граници се приближават една към друга и това ще продължи до събирането им което ще е след n стъпки.

Цялостната сложност е O(n2) – квадратично време с приблизителен брой стъпки n \* n (n2).

1. **What is the expected running time of the following C# code? Explain why. Assume the input matrix has size of n \* m.**

**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;**

**}**

Външният цикъл ще се изпълни n пъти – колкото редове има матрицата. До вътрешния цикъл ще се стига само когато елемента от матрицата на съответния ред и 0ева колона е четен. Това означава че изпълнението изцяло зависи от елементите в матрицата. Може да разгледаме 3те случея. В най-лошия всеки елемент ще е четен и ще се стига до цикъла всеки път. Това би означавало цялостна сложност от n \* m. В най-добрия случей всеки елемент би бил нечетен и никога няма да се стигне до вътрешния цикъл – тогава сложността ще е линейна. В средния случей всеки втори път ще се стига до вътрешния цикъл – ако всеки втори елемент е четен. Това прави сложност от n \* m/2. Като цяло този алгоритъм е много невероятно да е само с нечетни елементи в матрицата което прави сложността O(n2).

1. **\* What is the expected running time of the following C# code? Explain why. Assume the input matrix has size of n \* m.**

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

**{**

**long sum = 0;**

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

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

**if (row + 1 < matrix.GetLength(0))**

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

**return sum;**

**}**

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

Считаме, че задачата не е правилна и сменяме GetLength от 0 на 1 на двете места (иначе се хвърля exception при разлика в редовете и колоните). Външния for цикъл ще се изпълни n пъти (колкото колони има матрицата). Вътрешния цикъл ще се изпълни всеки път когато конкретния ред + 1 е по-малък от n(редовете на матрицата). Тоест цялостната сложност е О(n\*m) – квадратично време с приблизителен брой стъпки n \* m.