Count pairs from two sorted matrices with given sum

Given two sorted matrices **mat1** and **mat2** of size **n x n** of distinct elements. Given a value **x**. The problem is to count all pairs from both matrices whose sum is equal to **x**.

Note: The pair has an element from each matrix. Matrices are strictly sorted which means that matrices are sorted in a way such that all elements in a row are sorted in increasing order and for row 'i', where 1 <= i <= n-1, first element of row 'i' is greater than the last element of row 'i-1'.

```
Input : mat1[][] = { {1, 5, 6},
                     {8, 10, 11},
                    {15, 16, 18} }

    mat2[][] = { {2, 4, 7},
                 {9, 10, 12},
                 {13, 16, 20} }
       x = 21
Output : 4
The pairs are:
(1, 20), (5, 16), (8, 13) and (11, 10).
```

```
def searchInMatrix(matrix, target):
    n = len(matrix)
    i = 0
    j = n - 1
    while i >= 0 and i < n and j >= 0 and j < n:
        val = matrix[i][j]
        if val == target:
            return val
        elif val > target:
            j = j - 1
        else:
            i = i + 1
    return None


def pairsINMatrix(matrix, matrix2, target):
    ans = []
    for i in range(len(matrix)):
```

```
        for j in range(len(matrix)):
            temp = target - matrix[i][j]
            val = searchInMatrix(matrix2, temp)
            if val is not None:
                ans.append((matrix[i][j], val))
    return ans


matrix = [[1, 5, 6],
          [8, 10, 11],
          [15, 16, 18]]
matrix2 = [[2, 4, 7],
           [9, 10, 12],
           [13, 16, 20]]
target = 21

print(pairsINMatrix(matrix, matrix2, target))
```