Count the number of possible triangles

Given an unsorted array **arr[]** of **n** positive integers. Find the number of triangles that can be formed with three different array elements as lengths of three sides of triangles.

Example 1:

```
Input:
n = 3
arr[] = {3, 5, 4}
Output:
1
Explanation:
A triangle is possible
with all the elements 5, 3 and 4.
```

Example 2:

```
Input:
n = 5
arr[] = {6, 4, 9, 7, 8}
Output:
10
Explanation:
There are 10 triangles
possible  with the given elements like
(6,4,9), (6,7,8),...
```

Your Task:
This is a function problem. You only need to complete the function **findNumberOfTriangles**()** that takes **arr[]** and **n** as input parameters and returns the count of total possible triangles.

Expected Time Complexity: $O(n^2)$.
Expected Space Complexity: O(1).
Constraints:

$3 <= n <= 10^3$

$1 <= arr[i] <= 10^3$

```
class Solution:
    #Function to count the number of possible triangles.
    def findNumberOfTriangles(self, arr, n):
        #code here
        arr.sort(reverse = True)
```

```
        count = 0
        n = len(arr)-1
        for i in range(len(arr)):
            target = arr[i]
            j = i+1
            k = n
            while j<k:
                temp = arr[j]+arr[k]
                if temp>target:
                    count+=k-j
                    j = j+1
                else:
                    k-=1

        return count
```