

Counting Triangles

Given a list of **N** triangles with integer side lengths, determine how many different triangles there are. Two triangles are considered to be the same if they can both be placed on the plane such that their vertices occupy exactly the same three points.

Signature

```
int countDistinctTriangles(ArrayList<Sides> arr)
or
int countDistinctTriangles(int[][] arr)
```

Input

In some languages, `arr` is an **N** \times 3 array where `arr[i]` is a length-3 array that contains the side lengths of the *i*th triangle. In other languages, `arr` is a list of structs/objects that each represent a single triangle with side lengths *a*, *b*, and *c*.

It's guaranteed that all triplets of side lengths represent real triangles.

All side lengths are in the range [1, 1,000,000,000]
 $1 \leq N \leq 1,000,000$

Output

Return the number of distinct triangles in the list.

Example 1

```
arr = [[2, 2, 3], [3, 2, 2], [2, 5, 6]]
output = 2
```

The first two triangles are the same, so there are only 2 distinct triangles.

Example 2

```
arr = [[8, 4, 6], [100, 101, 102], [84, 93, 173]]
output = 3
```

All of these triangles are distinct.

Example 3

```
arr = [[5, 8, 9], [5, 9, 8], [9, 5, 8], [9, 8, 5], [8, 9, 5], [8, 5, 9]]
output = 1
```

All of these triangles are the same.