Kick Start 2017 - Round E

Analysis: Trapezoid Counting

Trapezoid Counting: Analysis

Small dataset

There are at most 50 sticks, which means we can enumerate all different sets of four sticks directly.

For a set of sticks to form the four sides of an isosceles trapezoid, it must meet all of the following conditions:

- There must be a pair of sticks with equal length. Call this length C.
- The remaining two sticks must have unequal lengths. Call the shorter of those lengths A, and the longer one B.
- The four sticks must be able to actually connect to form an isosceles trapezoid, so we need to obey an isosceles trapezoid equivalent of the triangle inequality: $B < A + 2 \times C$.

Then we just need to count how many different sets of four sticks meet these conditions.

Large dataset

First, we can create a Length array with all the unique length values, and a Count array with the counts of each of those values.
For example, for sticks with lengths [3, 4, 1, 4, 2, 6, 3, 1, 3], we would get the Length array [1, 2, 3, 4, 6] and the Count array [2, 1, 3, 2, 1].

Consider the inequality above: $B < A + 2 \times C$.
There are two possible situations in which valid isosceles trapezoids can be formed:

- A is equal to C or B is equal to C.
- A, B, and C are all different values. (Recall that we cannot have A equal B, or else the shape would not meet the problem's definition of an isosceles trapezoid.)

For the first situation, we consider all possible values C = Length[i] such that Count[i] ≥ 3.
For each of these, we have to count how many sticks have length less than $3 \times C$ but not equal to C.
We can do this quickly by using a prefix sum of the Count array (which we only need to create once). Then we add that number, multiplied by (Count[i] choose 3), to our answer.

For the second situation, we consider all possible values C = Length[i] such that Count[i] ≥ 2.
For each of these, we consider all A = Length[j] (with $i \neq j$).
For each of those (C, A) pairs, we need to count how many sticks have length B such that $B \neq C$, $A < B$, and $B < A + 2 \times C$.
The number of valid B is the prefix sum of Count up to $A + 2 \times C$, minus the prefix sum of Count up to A, possibly minus the Count of C if it is in that range.
Then we add that number, multiplied by (Count[i] choose 2), to our answer.

This method avoids double-counting any sets, and it runs in $O(\mathbf{N}^2)$ time, which is easily fast enough for $\mathbf{N} \leq 5000$.