First of all, the description of the question is very long, which is good for practicing quickly understanding the questions.

The main observation needed to solve this problem is that the inchworm only eats at rest. The inchworm eats a leaf at the beginning of the branch. Then she inches forward the same distance as before, and repeats this routine until she has reached or passed the end of the branch and leaves are spaced at uniform intervals.

So, I use the For Loop. Firstly, to set a variable before the loop starts (int i = 0), to define the condition for the loop to run (i must be less than and equal branch). If the condition is true, the loop will start over again, if it is false, the loop will end. And, i equal i plus the distance traveled by the inchworm between rests (i=i+rest) each time the code block in the loop has been executed. If the distance between i and the leaf is 0, the number of leaves eaten increases by 1.

Inchworm: 105 out of 105.

After the previous practice and examination, I found that it took me some time to understand the question, so I used the opportunity of practice to choose a slightly more complicated question. And I want to practice using iterators.

VolumeGuess: 48 out of 48.
Java Classes: HashMap (put, get, values), ArrayList (size, get)
Interfaces: Collection, Iterator, Set, List, Map(entrySet)
Methods: Collections.sort

The important thing is that queries will contain no duplicates, no pair of boxes will be compared twice and no boxes have same volume. To compare every pair of boxes and get the volume of the smaller box. Return the size of ithBox.

Traversing through an ArrayList is easy, however, traversing through a Set container is tricky. So, using the Iterator interface, let the various containers override hasNext() and next() methods themselves, respectively. Meanwhile, I used the Map.Entry interface to traverse the Map.

There are some challenges in understanding and using iterators. And the core methods of the Iterator interface, next() or hasNext(), depend on the current iteration location of the Iterator. Be distinct from the interface Iterable.

The benefit of practicing Iterator iterators is that I can iterate over collections in the same way, without having to worry about the internal results and structure of the collection.