HW 0 Problem 4

Friday, June 5, 2020 9:55 PM

1.

Ball.java had 3 areas where it was messed up. Two member functions and the constructor. The member functions were not returning this.attribute so I made them do so. The code still didn't work so I stepped through with the debugger and found that the ball wasn't being made properly. I fixed the constructor by setting the lines to this.attribute = attribute.

2.

For add() I just the built Java method to add and used conditionals to match the desired output

For remove() I just the built Java method to add and used conditionals to match the desired output

For getVolume() I used the built in Java iterator to run through and sum all the balls volume

For size() I just used the built Java method for sets.

For differentColors() I used a hash set to store colors and returned it's size at the end because set's already take care of duplicates by design.

For areSameColor() I used the same logic as different colors and checked to see if the size was greater than 1 hence implying more than one color in the container.

For clear() I just used the built Java method for sets.

For contains() I just used the built Java method for sets.

Which approach do you think is the better one? Why? Include your answer in hw0_problem4.pdf.

The better solution is to keep track of the volumes during the adding and removing of the balls because those are O(1) where the other implementation is O(n) each time it is called.

3.

There are many ways to implement getBallsFromSmallest(). Briefly describe at least two different ways. Your answers should differ in the implementation of Box, not in lower-level implementation (for example, using an insertion sort instead of a selection sort is a lower-level implementation because it does not affect how Box is implemented). Hint: think about different places in the Box class where you could add code to achieve the desired functionality.

Run through the list $O(n^2)$ times in order to find the smallest element each time and store that in an array. Return an iterator to that array.

During the add of more balls use BST to store least to greatest order and store an iterator to that. Be sure to keep track of duplicates. Return that iterator when desired.

Which of the above ways do you think is the best? Why?

The second one is better because a lot of the time memory is not a huge problem. It's efficiently sorting through that memory is the tough part. Smaller runtime O() notation is usually better. You can easily add another 1TB of memory but you cannot easily make it run polynomial times better.