The bin packing problem is a problem of storing items in boxes based purely on their volume. Items are represented as an unchanging length, width, and height, as are the boxes. This is considered an np problem, simply put, this means there is no one best solution, because the pieces making up the problem will be different each time the problem needs a solution.

My algorithm is an any-fit algorithm, using an online system, and keeping all boxes open. What this means is that it will place the item in the first spot it finds that the item can fit. It being an online algorithm means that it will get one item at a time and place it, and that item can no longer be moved within the system. Keeping all boxes open means that the algorithm will check each space in every box, instead of closing each box once one item doesn’t fit.

The main part of the logic in my algorithm is where it gets all the values for a given item. It attempts to place the item in each position in every box at each of the six possible rotations and gets all of the 3D coordinate locations that the item would take up in that box, location, and rotation. Once it has all the values, it will go through and either return true or false for each of those values, depending on whether it intersects another item or goes outside the bounds of the box. After that, it will go through all of the values and find the first value that the item can fit into and places it there.

My algorithm is set up to someday implement machine learning. This will be done by following the same steps as above but returning a value instead of a true or false. The value would be negative one if the item does not fit, or it would be some value decided by a neural network for all other locations. It is hard to determine what those values would be, the idea is that the neural network would find connections and patterns to find the best spot for given objects.