Step 0:

The first step, we are simply sorting the points by the value of X. We use merge sort to do this. We do this because we have to split up the X and Y because it is impossible to compare x,y pairs, you have to compare just one or the other, not both.

Step 1:

This is our base case because we are using recursion to solve this problem. If the amount of points is <= to 3 we simply sort the points by the Y, then return the closest point using the bruteForce method. We can use brute force for 3 because it is a small amount of points to be compared.

Step 2:

We find the middle (median) of all X points and split it into two parts, left and right. We do this so that in part 3 we can divide and conquer.

Step 3:

We make a recursive call to this algorithm to divide and conquer. One call is the left half of the points and the other call is the right half of the points.

Step 4:

Now we get the distance from the first and second element for the left and right closest point. We then check if the left is closer to the left point then the right point is to the other right point. If it is, we know that the left point is closer, if it isn't we know the right point is closer. So we set the closest and dMin variable respectively.

Step 5:

We now merge sort based on the Y variables. To set us up for part 6.

Step 6:

Now that it is sorted, we can now get all points within the range of the closest point using the dMin variable. The way getPointsInRange method works is it checks all points within the range of left to right and checks if the x is between xmid - dMin and xmid + dMin. This calculator tells us whether or not the point is within the range we are searching for. Once that method is done, it returns a list of all points within the range. This will return a list of points that are as close or even closer then the points we compared.

Step 7:

Now we loop through all of the points and check the distance between points. This will ensure that we have the closest point. At the end we simply return the list of closest points.