6.006 Homework	**Problem set 7**	# 1 – Ghostbusters and Ghosts
Xola Ntumy		May 6, 2011
Collaborators: None		

(a) Since no three positions can be collinear if the Ghostbusters are in fixed positions there exists a line in the plane passing through one Ghostbuster and a ghost such that the number of Ghostbusters on one side of the line equals the number of ghosts on the same side. Such a line might pass through the Ghostbuster-ghost pair that both have the lowest or largest x or y-coordinates. This pair might be at the extremes of the plane and we know that no other ghost or Ghostbusters can exist on that line (otherwise three positions would be collinear). Therefore on one side of this line, there will be nothing and on the other side there will be the remaining pairs. This line will exist while the Ghostbusters are paired such that no streams are crossing.

To find such a line, we assume that the line exists and arbitrarily choose a Ghostbuster through which the line passes. Now we must find the ghost that pairs with the Ghostbuster in this line. To do this, we place each ghost in a binary search tree on basis of their distance away from the Ghostbuster. Now we search through the tree, for each ghost checking to see whether the difference between the number of Ghostbusters and ghosts on each side of the line drawn from the pair of that ghost and our arbitrary Ghostbuster is equal to 0. If it is, we have found our line, if it is negative we move to the left subtree, if it is positive to the right.

The runtime for this algorithm is O(n log n) simply because it takes O (log (n)) time to search the binary tree and O(n) time for each Ghostbuster we try.

(b) To pair each Ghostbuster with a ghost run the above algorithm n times. The reason why this would work is that if the aforementioned line is found for each ghostbuster, we can be assured that the ghost through which the line passes will allow pairings such that no streams cross. This would give a runtime of O(n*n log n).

6.006 Homework	**Problem set 7**	# 2 – Three is company
Xola Ntumy		May 6, 2011
Collaborators: None		

An algorithm to find a compatible roommate in O(m to the power log 3) time is as follows. First, represent A, B and C as a sequence of 0-1 coefficients of a polynomial. By multiplying the polynomials A and B, the exponents would be added up. If the product contains any of the terms from the polynomial C, then this would mean that there is some a in A, b in B and c in C for which a + b = c.

Multiplying the polynomials can be done in O(m to the power log 3) time using Karatsuba's algorithm for integer multiplication.

In this algorithm, the two integers are split into high (Ah, Bh) and low (Al, Bl) parts, each m/2 digits long.

Algorithm for mult(A,B):

a = mult(Ah, Bh)

e=mult(Ah, Bl) + mult(Al, Bh)

d= mult(Al, Bl)

return add(lshift(a,m/2), lshift(b, m/2), e)