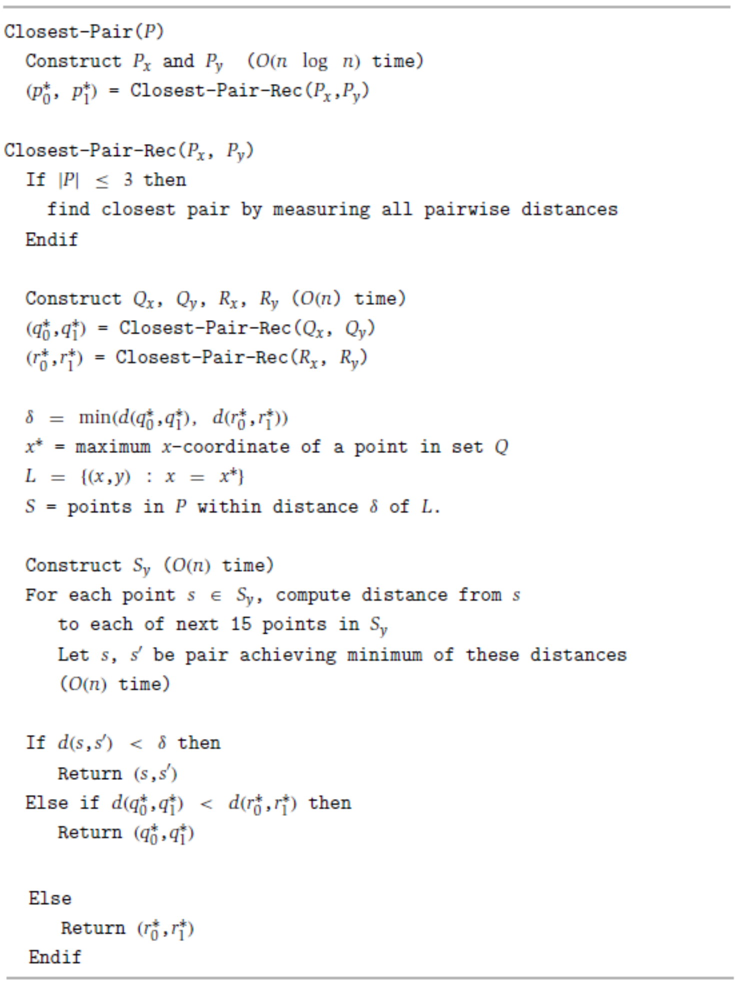
## Homework 5 Instructions

The goal of this homework is to find the closest pair of points using the algorithm in the *Algorithm Design* book presented on page 230. The pseudo code is as follows:

Closest Pair of Points



```
Closest-Pair($P$)
  Construct $P_x$ and $P_y$  ($O(n \log n)$ time)
  ($p_0^*$, $p_1^*$) = Closest-Pair-Rec($P_x$,$P_y$)

Closest-Pair-Rec($P_x$, $P_y$)
  If $|P| \leq 3$ then
    find closest pair by measuring all pairwise distances
  Endif

  Construct $Q_x$, $Q_y$, $R_x$, $R_y$ ($O(n)$ time)
  ($q_0^*$,$q_1^*$) = Closest-Pair-Rec($Q_x$, $Q_y$)
  ($r_0^*$,$r_1^*$) = Closest-Pair-Rec($R_x$, $R_y$)

  $\delta = \min(d(q_0^*,q_1^*),\ d(r_0^*,r_1^*))$
  $x^*$ = maximum $x$-coordinate of a point in set $Q$
  $L = \{(x,y) : x = x^*\}$
  $S$ = points in $P$ within distance $\delta$ of $L$.

  Construct $S_y$ ($O(n)$ time)
  For each point $s \in S_y$, compute distance from $s$
     to each of next 15 points in $S_y$
     Let $s$, $s'$ be pair achieving minimum of these distances
     ($O(n)$ time)

  If $d(s,s') < \delta$ then
     Return ($s$,$s'$)
  Else if $d(q_0^*,q_1^*) < d(r_0^*,r_1^*)$ then
     Return ($q_0^*$,$q_1^*$)

  Else
     Return ($r_0^*$,$r_1^*$)
  Endif
```

1. Create a python file named “closest_pair_id”, with id corresponding to your student id number. For example, if my student id number is 1234567, I would create a python file named “closest_pair_1234567”

2. Your program should be able to run from the console using the command

python closest_pair_1234567.py input.txt > output.txt where the arguments are your program and the input file. ***Your program should be able to write to an output file.*** However, while testing, you may print to stdout to check your results for debugging purposes. I will be using Python 3.6 to grade your assignments.

3. Your program should read in an input text file that contains a list of points on a graph. An example of the input file is given below:

   ```
   3 -4
   -1 -1
   2 0
   -5 -2
   ```

   Here we see that the points on the graph are (3, -4), (-1 -1), (2, 0), and (-5, -2).

4. The program **closest_pair** should take in the points and find the closest pair of points on the graph. For the points given above, the **closest_pair** program would output the following:

   ```
   -1 -1
   2 0
   ```

5. You can further test your program by creating input text files following the format of the sample input text files provided.

6. Please submit your assignment in the appropriate location on Blackboard. If you get stuck or have any questions, please feel free to email me at romoore@ku.edu. I will try to get back to you as quickly as possible. It is better to contact me during the week, as I am less responsive on the weekend because, well, it is the weekend. This does not mean that I will not respond to your emails on the weekend, it only means that the response time will be longer than it would be during the week. However, I will do the best I can to respond as quickly as possible.