**Project 3 Part 2 (intermediary recursive approach)**

[Project 3 Part 2 (intermediary recursive approach)](https://fcps.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_1516674_1&content_id=_43601203_1&mode=reset)

**Part 2:** create a method part2() that  
 Create and submit an application that solves the closest-pair problem in the unit square using the "brute force" approach discussed in class as well as the preliminary recursive algorithm.

For the recursive approach you must use at least one vector to store the points to have O(1) access to any element. Also you must use at least one iterator to traverse the elements of the vector.

Before you start the preliminary recursive call you should read the points from the file points.txt into a vector(this part should not be timed).

Preliminary recursive algorithm:

 a) Sort the vector based on the x coordinate; O(nlogn). This step is outside the recursive method and should be done only once. Next steps describe the recursive method.

 b) Divide the vector into 2 parts and recur for each part (logical division preferred since is faster); O(1) since is sorted

 c) If there are 3 or 2 points just simply return the minimum distance since it can be done in constant time O(1)

 d) When the 2 recursive calls return with 2 minimum distances (let's say d1 and d2), calculate d = min(d1,d2)

 e) create a strip of distance d to the left of the middle and d to the right of middle (middle value)

 f) for each point on the left of the strip calculate the distance to each point on the right side of the strip and if you find a smaller distance than d, then update the d. O(n^2) since it may have almost all points inside the strip (so you brute force the strip)

 g) return the minimum distance you obtained and the 2 points that have that distance

In the main you should:

a) call part1()

b) call part2()

c) display on the screen and in the results.txt the 2 points and minimum distance obtained for both approaches also the time to complete each approach (you may do this either by creating global variables or by making both part1 and part2 to return some result)

Please turn in to Mr. Jurj a printout of the following file filled:

[Project 3 Part 2.docx](https://fcps.blackboard.com/bbcswebdav/pid-44340706-dt-content-rid-48611251_2/xid-48611251_2) [Project 3 Part 2.docx - Alternative Formats](https://fcps.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_1516674_1&content_id=_43601203_1&mode=reset)

Obs:

- when you are timing the bruteforce do not include the time to create the list, just the time to find the 2 closest points

- when you are timing the preliminary recursive method do not include the time to create the vector, but do include the time to sort the vector and to find the 2 closest points. (yes the time to sort is a price you pay for this approach , is the overhead)

- when you compare the 2 methods make sure you test also for more points (go up to 1000, or even more if you wish)

- testing on terminals is faster since some of the machines have gpu's and stronger processors