**Project 3 Part 4 (randomized approach average O(n))**

Name: \_Riya Dev\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Period: \_5\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_11/17/2021\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Is your lab name l034?(lowercase L followed by digits 034) \_yes\_\_\_\_\_\_\_\_\_\_

Did you created a class to store a point? \_yes\_\_\_\_\_\_\_\_\_\_

Did you use a vector to store the points you read? \_yes\_\_\_\_\_\_\_\_\_\_

Does your main contain only 2 calls of: part3() and part4() (NO part1/2!!)? \_yes\_\_\_\_\_\_\_\_\_\_

(in main you may also have the part to display results for the 2 methods and them also in the txt file)

Obs: the part3 and part4 may have a return type like the time it took to complete the algorithm, the minimum distance, the points for the minimum distance, or you may create global variables for these.

Did you use an unordered\_map for your dictionary? \_yes\_\_\_\_\_\_\_\_\_\_

Did you implement the Knuth algorithm to randomize the points? \_yes\_\_\_\_\_\_\_\_\_\_

1. **Paste here a clear picture of the graph that compares the running times of the “full recursive” algorithm and “randomized” algorithm versus number of points. (use 2 different colors for the 2 graphs, colors that can be visible even if you print in black and white). Each point on this graph should be an average of several runs for that size:**
2. **Paste here the content of the results.txt when you run your lab on the content of the file points10k.txt and points100k.dat**

**For 10k:**

0.12286165147159841537494 (0.95857070123659116056558 , 0.61877366956345547510665) (0.98999999999999999111822 , 0.50000000000000000000000)

duration: 0.00000000000000000000000 seconds

0.12286165147159841537494 (0.98999999999999999111822 , 0.50000000000000000000000) (0.95857070123659116056558 , 0.61877366956345547510665)

duration: 0.00000000000000000000000 seconds

**For 100k:**