Instructions

Q1.   We discussed two versions of the 3-sum problem: A "naive" implementation (O(N^3)) and a "sophisticated" implementation (O(N^2 lg N)). Implement these algorithms. Your implementation should be able to read data in from regular data/text file with each entry on a separate line. Using Data provided under resource (hw1-1.data.zip) to determine the run time cost of your implementations as function of input
data size. Plot and analyze (discuss) your data.

Q2. We discussed the Union-Find algorithm in class. Implement the three versions: (i) Quick Find, (ii) Quick Union, and (iii) Quick Union with Weight Balancing. Using Data provided here (hw1-2.data.zip, under resources) determine the run time cost of your
implementation (as a function of input data size). Plot and analyze your data. Note: The maximum value of a point label is 8192 for all the different input data set. This implies there could in principle be approximately 8192 x 8192 connections. Each line of the input data set contains an integer pair (p, q) which implies that p is connected to q.

Recall: UF algorithm should

// read in a sequence of pairs of integers (each in the range 1 to N) where N=8192

// calling find() for each pair: If the members of the pair are not already connected

// call union() and print the pair.

Q3. Recall the definition of "Big Oh" (where F(N) is said to be in O(g(N), when F(N) < c (g(N)), for N > Nc) . Estimate the value of Nc for both Q1 and Q2. More important than the specific value, is the process and reasoning your employ.

Q4: Farthest Pair (1 Dimension): Write a program that, given an array a[] of N double values, find a farthest pair: two values whose difference is no smaller than the difference of any other pair (in absolute value). The running time of the program should be LINEAR IN THE WORST CASE.

Q5. Faster-est-ist 3-sum: Develop an implementation that uses a linear algorithm to count the number of pairs that sum to zero after the array is sorted (instead of the binary-search based linearithmic algorithm). Use the ideas to develop a quadratic algorithm for the 3-sum problem.