CS 341, Spring 2022 T. Brown

PROGRAMMING ASSIGNMENT 1

DUE: Wednesday June 15, 11:59 PM. DO NOT COPY. ACKNOWLEDGE YOUR SOURCES.

Please read `http://www.student.cs.uwaterloo.ca/~cs341` for general instructions and policies.

Closest Pair Problem

Problem Statement

In this assignment, you are asked to implement a Divide and Conquer algorithm to solve Closest Pair Problem from Lecture Notes. That is, given a point set P with n Points, you want to find two points that have the shortest distance among all the other pairs; you should return both the pair and their distance.

Input Format

In the first line of input you are given the integer t. t represents the number of point sets you are given (basically the number of *closest pair problem* your program should solve. In the following lines, you are going to be given t instance of *closest pair problem*.

In each instance, first you are given the integer n in the first line. n denotes the number of points in the point set. Then in the next n lines of each instance, the coordinates of the each point is given. Specifically, each line would contain two floats (with at most two decimal places) representing x and y-coordinates of each point in order. The points in each point set are numbered in the same order they are given in the input.

Output Format

For each instance of the problem you need to output both the closest pair and their distance. That is, for each instance first line of output should be the distance between closest pair and second line consists of two integer representing the points in the closest pair.

This means your output consists of $2t$ lines, line $2i - 1$ is the distance between closest pair in instance i of problem and line $2i$ contains two integer that shows two points having shortest distance in instance i of the problem.

Also, note that the distances you are returning should be rounded to 4 decimal places.

Constraints

For all test cases, $1 \leq t \leq 50$, $2 \leq n \leq 20000$, $0.0 \leq$ points coordinations ≤ 100000.0.

10 Points of this assignment needs an efficient implementation for the problem.

Sample Input 1

```
1
4
2.00 5.00
4.00 2.00
7.00 7.00
8.00 1.00
```

Sample Output 1

```
3.6056
0 1
```

Sample Input 2

```
2
2
15.00 20.00
18.00 16.00
10
41.68 294.82
25.25 975.95
409.14 855.64
739.81 953.23
553.06 408.78
213.01 117.66
585.40 541.84
740.86 224.80
72.69 668.53
243.42 840.93
```

Sample Output 2

```
5.0
0 1
136.9337
4 6
```

Submission Instructions

- Submit your solution on Marmoset.
- You can choose to code in either C++ or Python.
- Name your program `prog1.cpp/prog1.py`
- **Time limit**: 7 seconds (C++) / 22 seconds (Python) for each test case.

- Compilation command for C++: `g++ -std=c++14 prog1.cpp -O3 -o prog1`
- Execution command for Python: `python3 prog1.py`
- Read from standard input and write to standard output.
- There will be several test cases, worth a total of 100 points. The public tests are worth 10 points and the secret tests are worth 90 points. The public tests (input only, not the answer) will be made available under a separate file.
- We will take the submission with the highest score. Please, however, refrain from excessive submissions.
- General collaboration policy applies. Please acknowledge your collaborator(s) by adding a comment in the beginning of your code.
- FAQ and updates will be posted on Piazza when necessary