**Computer Science & Computer Engineering 371  
Programming Assignment #2 -- Shortest Path Around**

**Objectives:**

* To implement the Convex Hull Divide and Conquer algorithm.
* To practice developing pseudo code from a description of ideas.
* To practice using the divide and conquer algorithmic design technique.
* To practice using recursion to implement divide and conquer.

**Overview:**

There is an area in the two-dimensional Euclidean plane defined by a set of points P1(x1, y1), ... Pn(xn, yn). This set of points defines a convex region that must not be entered. You are also given a pair of points A(xa, ya) and B(xb, yb). Find the shortest path from A to B that does not enter the region defined by the convex hull of the points P1, ..., Pn. Your solution must use the text's recursive divide and conquer algorithm to find the convex hull of P1, ..., Pn.

**Program Input**

Your program must read from a file called pairs.txt. The file is in the following format. The first line contains a single integer, which is the number of problems (0 < nProbs < 1000) that your program should solve. Following that, there is a set of data for each of the problems. The data for a single problem has the following format. The first line contains four numbers (xa ya xb yb) for the points A and B. Then there is a single number on a line by itself indicating the number of points (0 < np < 100000). Following that there are np lines, each line represents a single point in the set with two decimal numbers separated by spaces, which are the x and y coordinates of the point.

**Program Output**

All decimal values should have two decimal places. Each problem's output should start with a line (with a blank line after the preceding problem, if any) that says "Problem #i" where i is the problem number.

On the next line should be the text "Convex Hull:". On the following lines there should be the points of the convex hull in a clockwise path order (the point on the line following must be adjacent to the current point), one point per line.

After the lines with the convex hull points there should be a line with "Shortest Path:". On the following line give the starting point for the shortest path problem (point A above). On subsequent lines with one point per line should be the convex hull points of the shortest path. On the last line is the point B followed by the word "distance: " and the total length of the path from A to B.

Problems should be separated by a blank line.

In general format your output to match the example shown below.

**Example Input**

Your program should treat all x y coordinate pairs as doubles.

2

0 0 5 5

7

1 1

2 2

1.5 1.5

1.2 1.1

1 2

2 1

1.7 1.5

1.5 0 1.5 3

7

1 1

2 2

1.5 1.5

1.2 1.1

1 2

2 1

1.7 1.5

**Example Ouput**

Problem #1:

CH: (1.00, 1.00 )

(1.00, 2.00 )

(2.00, 2.00 )

(2.00, 1.00 )

PATH:

A: (0.00, 0.00)

(1.00, 2.00)

B: (5.00, 5.00) distance 7.24

Problem #2:

CH:

(1.00, 1.00 )

(1.00, 2.00 )

(2.00, 2.00 )

(2.00, 1.00 )

PATH:

A: (1.50, 0.00)

(1.00, 1.00)

(1.00, 2.00)

B: (1.50, 3.00) distance 3.24

**Program Design**

Use the algorithm described in the book in Chapter 5 Section 5. There is no neat pseudo code given so part of the problem will be for you to develop that yourself. (We did a lot of it in class.)

**Hints**

1. Use a stable JAVA sort to organize the points.
2. Use a JAVA Collections object to hold your data.
3. You will need two Comparators to be able to sort both by x and by y.
4. Creating a *Point* class is a good idea.
5. Beware round-off error in decimal comparisons.

**On the due date at the beginning of class turn in:**

* A hard copy of your source code (see the programs web page for additional requirements).
* A hard copy of your tests runs showing output correctness for your own data.
* A hard copy of the Javadocs for your classes, or the equivalent if you are using a language other than Java.
* Suggestions for improving this lab assignment (optional).
* Your source (.java ) files via [Submit](https://hub.cs.plu.edu/submit/).