

# Points, Points, and More Points

— A programming project

## 1 Introduction

Egyptian and Greek geometry were masterpieces of applied mathematics. The original motivation for geometric problems was the need to tax lands accurately and fairly and to erect buildings. As often happens, the mathematics that developed had permanence and significance that far transcend the Pharaoh's original revenue problem, for Geometry is at the heart of mathematical thinking. It is a field in which intuition abounds and new discoveries are within the compass (so to speak) of nonspecialists.

Computational geometry is the branch of computer science that studies algorithms for solving geometric problems. In modern engineering and mathematics, computational geometry has applications in, among other fields, computer graphics, robotics, VLSI design, computer-aided design, and statistics. The input to a computational-geometry problem is typically a description of a set of geometric objects, such as a set of points, a set of line segments, or the vertices of a polygon in counterclockwise order. The output is often a response to a query about the objects, such as whether any of the lines intersect, or perhaps a new geometric object, such as the convex hull (the smallest enclosing convex polygon) of the set of points.

In the project, you are asked to implement classes that represent geometric objects, such as points, point pairs, point pair sets, and point sets. The problem you need to solve eventually is to find the closest pair of points among a given point set on the 2-dimensional plane. The project involves class implementation, dynamic arrays, linked lists, sorting, divide-and-conquer, and random number generation. The goal of this project is to obtain first-hand experience on how good algorithm design can speed up problem solving, as well as to review some of the important programming aspects of C++.

## 2 The Point Class and Point Pair Class

The first two class you are asked to implement are the point class and Point Pair Class.

A point on a 2-D plane is defined by the  $x$  and  $y$  coordinates, which are real numbers in principle, but are assumed to be positive integers in a certain range in this project. The header file for the point class `Point.h` can be downloaded from the project page.

A point pair on a 2-D plane contains two 2-D points. The header file for the point pair class `PointPair.h` can be downloaded from the project page.

To receive credits for this part of the project submit your `Point.cpp` and `PointPair.cpp` by the deadline, which is the midnight of the due date.

*To be continued ...*