ES 103

Term 1, 2015-16

Assignment 6

A number of applications require the ability to model and run computations on points (locations) in different dimensions – on a line, a plane, in 3 space, etc. Mathematically, we deal with these different objects in an elegant, uniform way, using concepts like distance, interval, containment, etc. Further, the concepts largely hold whether we use real numbers or integer coordinates. However, implementing these in software is comparatively messy and time-consuming.

Part 1.

Can we design a set of classes and functions in C++ that simplifies this process and provides a convenient abstraction of this mathematical concept? Specifically, we need the following:

1. Defining a “point” in *n* space, where n >= 1
2. Defining the following operations on these objects:
    a. Distance from the origin
    b. Distance between two points
    c. Find the closest point to a given point from a set of points
    d. The “rectangular bounding interval” of a set of points. This can itself be modelled as the min and max (or “lower left” and “upper right”) points of the interval that bounds the points.
    e. Test if a point intersects (or is contained in) an interval
    f. Test for the intersection of two intervals
    g. Ordering a set of points based on:
        i. Their distance from a given point
        ii. “lexicographic” ordering: sort by coordinates of first dimension, then the second dimension, then the third, and so on..
3. All classes and methods should be “generic” (and parameterized)
    a. Can work with any dimensional space (line, plane, 3-space, 4-space, etc)
    b. Can work on real number or integer coordinate systems (and hopefully in any other space where certain measures are defined. What would those be?)

You should define the Point class and its methods and related functions explicitly. You can use the C++ Standard Lib for containers, iterators, algorithms, etc. inside these classes and functions – in fact, you are encouraged to do so.

The program should be demonstrated on data sets for points in 1, 2, 3, and 4 D space.

Part 2.

Can you use these classes and functions to re-implement the Moving Discs program? You can use either your solution or the sample solution provided.

Can this game now be modified to become the “Floating Spheres” game – same rules as before except we are dealing with spherical balls moving around in an empty room?