

CS 499 (Fall 2006)- Assignment 7

Due: Thursday, 10/19/2006

- (1) Write a program that is given a set of polygons as an input, and computes the total area covered by those polygons. All polygons will be simple (i.e., not self-intersecting), but multiple polygons could overlap each other. If multiple polygons cover the same area, it is supposed to be counted only once. (Problem 2 suggests an inexact way to compute this quantity. Implementing that method instead of an exact one will only be worth partial credit.)
- (2) A simpler, but inexact, way to solve the problem is the following: find the smallest axis-aligned rectangle containing all polygons. Now, generate a lot of points in that rectangle independently and uniformly at random. Count the percentage of points that ended up in at least one of your polygons (using the Point-in-Polygon algorithm). From this percentage, you can then infer the total area of the polygons.
 1. For what type of inputs will this method be most inexact (in terms of relative error in estimating the correct area)?
 2. To improve this method somewhat, you could generate points not inside the smallest rectangle, but inside the convex hull of all the polygons. What do bad example look like now? What additional difficulties do you run into?