Computer Science 715 S2C (2007)

Assignment 2 (individual assignment)

Due date September 20, 2007

Goals

In this assignment we want you to write either a simple graph drawing algorithm or an efficient algorithm to compute convex hulls. In both cases, the PostScript rendering language will be used to display your results.

You only need to do one of the following parts. (If you do both then your grade will be the higher of the two scores.)

Problem 1: Spring Graph Drawing Algorithm

1. Write an algorithm that reads in a graph (input format described below) and computes a nice spring layout.
2. You program should be able handle up to 50 vertices.
3. Generate and submit a small symmetric (about 10 vertices), medium planar (about 20 vertices) and large (about 30 vertices) test graphs.
4. Submit best drawings obtained by your algorithm for your three test cases.
5. Provide a short 1-page text description on how to use your program.

Input for this problem consists of an undirected graph. Each graph is represented by an adjacency list. The first line is an integer n indicating the order of the graph. This is followed by n white space separated lists of adjacencies for vertices labeled 0 to $n-1$.

Problem 2: Computing Convex Hull

1. Write an efficient $O(n \log n)$ algorithm that reads in a set of points (input format described below) and computes their convex hull.
2. You program should be able process at least 1000 points.
3. Generate and submit one small, one medium and one large test cases (about 20, 100 and 500 points).
4. Submit hull drawings obtained by your algorithm for your three test cases.
5. Provide a short 1-page text description on how to use your program.

Input for this problem consists of a integer n, denoting the number of points, then starting on the next line a sequence of x and y coordinates of the points (all white space separated). Each x and y will be an integer between -1000 and 1000.

Submission and Due Date

Submit your source code, usage instructions, and test cases (and results) to the CompSci 715 assignment drop box on or before the due date. This assignment is worth 15% of your course marks.

Output Format

Output for both problems should be a rendering of the graph drawing or hull in Encapsulated PostScript format (e.g. see `http://www.tailrecursive.org/postscript/eps.html`). You may want to use (which is optional) the following template.

```
%!PS-Adobe-2.0
%%
%%Creator: Michael J. Dinneen
%%Title: CompSci 715 sample PostScript code for graphics
%%
%%BoundingBox: 60 60 540 230

% Sample postscript functions to draw vertices/points and edges/lines

/node { 6 0 360 arc gsave 1 setgray fill stroke grestore stroke } def
/point { 2 0 360 arc gsave 0 setgray fill stroke grestore stroke } def

/line { moveto lineto stroke } def
/edge { gsave 2 setlinewidth line grestore } def
/hull { gsave [4 2] 0 setdash line grestore } def

% Sample graph drawing example

/v0 { 100 100 } def
/v1 { 120 180 } def
/v2 { 200 200 } def
/v3 { 180 120 } def

v0 v1 edge v1 v2 edge v2 v3 edge v3 v0 edge v1 v3 edge
v0 node v1 node v2 node v3 node

% Sample convex hull example

/p0 { 300 100 } def
/p1 { 320 180 } def
/p2 { 380 200 } def
/p3 { 480 120 } def
/p4 { 350 140 } def
/p5 { 430 120 } def
/p6 { 430 140 } def
/p7 { 320 160 } def

p0 point p1 point p2 point p3 point p4 point p5 point p6 point p7 point
p0 p1 hull p1 p2 hull p2 p3 hull p3 p0 hull
```

This PostScript is rendered as: