Devin Hardy

CS 472

Assignment 4

// Devin Hardy

// CS472

#include <iostream>

#include <fstream>

#include <vector>

#include <utility>

#include <random>

using namespace std;

////////////////////////////

//Functions for problem 1 //

////////////////////////////

//Bool function for checking signs

//All points on right side of line

bool Signs(vector<int> &nums)

{

bool rSide, lSide;

rSide = true;

lSide = true;

for(int i = 0; i < nums.size(); i++)

{

if(nums[i] > 0)

lSide = false;

if(nums[i] < 0)

rSide = false;

}

if((lSide == false && rSide == true))

return 1;

return 0;

}

//Insert a reference to a vector of pairs and return a reference to a vector of pairs

void ConvexHull(vector<pair<int, int>> &points, vector<pair<int, int>> &hull)

{

int a=0, b=0, c=0, num=0;

pair<int, int> point1;

pair<int, int> point2;

vector<int> signs;

for(int i = 0; i < points.size(); i++)

{

for(int j = 0; j < points.size(); j++)

{

if(points[j] == points[i])

{

continue;

}

else

{

// a, b, c math

a = points[j].second - points[i].second;

b = points[i].first - points[j].first;

c = ((points[i].first \* points[j].second) - (points[i].second \* points[j].first));

}

for(int k = 0; k < points.size(); k++)

{

if(points[k] == points[i] && points[k] == points[j])

{

continue;

}

else

{

//signs math

num = ((a \* points[k].first) + (b \* points[k].second) - c);

signs.push\_back(num);

}

}

//check signs

//if signs == true

// add Pi and Pj to convex hull

if(Signs(signs))

{

point1.first = points[i].first;

point1.second = points[i].second;

point2.first = points[j].first;

point2.second = points[j].second;

hull.push\_back(point1);

hull.push\_back(point2);

}

// Clear signs vector

signs.clear();

}

}

return;

}

//////////////////////////

// Problem 2 Functions ///

//////////////////////////

// Random

// a = min, b = max

int RANDOM(int a, int b)

{

static random\_device dev;

mt19937 eng{dev()};

uniform\_int\_distribution<int> dist{a, b};

int num = dist(eng);

return num;

}

// Read from file to get points

void CreateObserve(ifstream& in, vector<pair<int,int>> &mapPoints)

{

int num1, num2;

pair<int, int> pnt;

char junk;

while(in >> num1)

{

in >> junk >> num2;

pnt.first = num1;

pnt.second = num2;

mapPoints.push\_back(pnt);

}

return;

}

int main()

{

vector<pair<int, int> > points1; // random points pairs 1

vector<pair<int, int> > hull1; // Convex Hull pairs 1

vector<pair<int, int> > points2; // random points pairs 2

vector<pair<int, int> > hull2; // Convex Hull pairs 2

// Problem 1

//quick check if works

//first enter data

pair<int, int> pnt1(1, 6);

points1.push\_back(pnt1);

pair<int, int> pnt2(4, 6);

points1.push\_back(pnt2);

pair<int, int> pnt3(5, 2);

points1.push\_back(pnt3);

pair<int, int> pnt4(5, 4);

points1.push\_back(pnt4);

pair<int, int> pnt5(5, 8);

points1.push\_back(pnt5);

pair<int, int> pnt6(6, 8);

points1.push\_back(pnt6);

pair<int, int> pnt7(6, 10);

points1.push\_back(pnt7);

pair<int, int> pnt8(8, 5);

points1.push\_back(pnt8);

pair<int, int> pnt9(8, 10);

points1.push\_back(pnt9);

pair<int, int> pnt10(10, 8);

points1.push\_back(pnt10);

// Find convex hull

ConvexHull(points1, hull1);

// Display the convex hull vector of pairs

cout << "Convex Hull vector\n";

for(int i=0;i<hull1.size();i++)

{

cout<< "(" << hull1[i].first << "," << hull1[i].second << ") - ";

i++;

cout<< "(" << hull1[i].first << "," << hull1[i].second << ")\n";

}

cout << "\n\n\n";

// Problem 2

//Create data file portion

ofstream Observe;

Observe.open("ObservedData.out");

for(int i = 0; i <= 100; i++)

{

Observe << RANDOM(0, 1000) << " , " << RANDOM(0, 1000) << endl;

}

Observe.close();

//Read data file into vector

ifstream Data;

int num;

Data.open("ObservedData.out");

CreateObserve(Data, points2);

ConvexHull(points2, hull2);

cout << "Convex Hull for Observed data." << endl;

for(int i=0;i<hull2.size();i++)

{

cout<< "(" << hull2[i].first << "," << hull2[i].second << ") - ";

i++;

cout<< "(" << hull2[i].first << "," << hull2[i].second << ")\n";

}

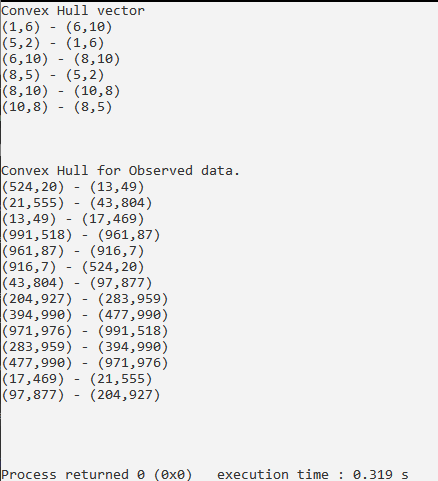
cout << "\n\n\n";

Data.close();

return 0;

}

Output:



ObservedData.out file contents:

730 , 562

630 , 925

226 , 227

782 , 319

326 , 207

578 , 455

872 , 618

524 , 20

763 , 103

809 , 465

718 , 900

434 , 108

642 , 655

21 , 555

476 , 299

949 , 690

292 , 76

948 , 354

683 , 128

175 , 759

949 , 88

651 , 533

598 , 782

521 , 560

305 , 356

179 , 866

618 , 587

749 , 957

665 , 567

538 , 24

304 , 170

749 , 903

755 , 708

801 , 425

13 , 49

368 , 664

729 , 74

513 , 276

612 , 665

138 , 119

991 , 518

961 , 87

615 , 561

744 , 913

116 , 572

946 , 371

842 , 640

916 , 7

753 , 295

43 , 804

435 , 884

204 , 927

131 , 278

394 , 990

207 , 329

741 , 746

759 , 461

971 , 976

292 , 863

138 , 344

794 , 840

696 , 87

556 , 472

766 , 409

210 , 442

345 , 496

814 , 170

390 , 213

211 , 572

283 , 959

477 , 990

149 , 859

632 , 213

223 , 780

157 , 688

587 , 768

559 , 735

757 , 607

80 , 439

676 , 222

715 , 857

177 , 294

685 , 136

17 , 469

820 , 747

787 , 406

853 , 846

237 , 75

741 , 954

330 , 250

419 , 795

603 , 881

907 , 46

924 , 566

462 , 126

967 , 269

104 , 142

431 , 643

212 , 497

760 , 874

97 , 877

Work Cited

[1] sleepijs, Skydiver, and modi123\_1, “Help needed with convex-hull homework. - C and C++: Dream.in.code,” *: Programming & Web Development Community*, 20-Nov-2017. [Online]. Available: https://www.dreamincode.net/forums/topic/407711-help-needed-with-convex-hull-homework/. [Accessed: 21-Feb-2022].