Міністерство освіти і науки України

Національний Технічний Університет

«Харківський Політехнічний Інститут»

Кафедра «Стратегічного Керування»

Лабораторна робота № 7

«ГЕОМЕТРИЧНІ АЛГОРИТМИ»

Перевірила: Мошко Є. О.

Виконав:

Харків, 2018

Лабораторна робота № 7

Тема: Геометричні Алгоритми

Завдання 1

Розробити програму, яка читає з клавіатури число N (1 < N < 256) та N пар дійсних чисел —координати точок на площині. Програма виконує один з алгоритмів згідно варіанту.

Варіанти завдань

(2 варіант)

Точки належать ламаній. Потрібно для кожної нової ланки вказати, чи перетинає вона будь-яку з попередніх.

Хід роботи

|  |
| --- |
| #include <iostream>  using namespace std;  class Broken{  private:  int\*\* arr;  int count=0;  int p1\_min = 0;  int p1\_max = 0;  int p2\_min = 0;  int p2\_max = 0;  public:  Broken(){}  Broken(int N){  arr = new int\*[N];  for (int i=0; i<N; i++){ // Creating new array of coordinates for each edge  arr[i] = new int[4];  }  arr[0]=NULL;  }  void put(int p1, int p2){  if (arr[0] == NULL){  arr[0] = new int[4];  arr[0][0] = 0;  arr[0][1] = 0;  arr[0][2] = p1;  arr[0][3] = p2;  }  else  {  arr[count][0] = arr[count-1][2];  arr[count][1] = arr[count-1][3];  arr[count][2] = p1;  arr[count][3] = p2;  if ((p1\_max>=min(arr[count][0],arr[count][2]))&&(max(arr[count][0],arr[count][2])>=p1\_min)&&(p2\_max>=min(arr[count][1],arr[count][3]))&&(max(arr[count][1],arr[count][3])>=p2\_min)){  cout << "Rectangulars cross" << endl;  cout << p1 << " " << p2 << ": " << p1\_min << " " << p1\_min << " / " << p1\_max << " " << p2\_max << endl;  for (int i=0; i<count-1; i++){  if ((max(arr[i][0],arr[i][2])>=min(arr[count][0], arr[count][2]))&&(max(arr[count][0],arr[count][2])>=min(arr[i][0],arr[i][2]))&&(max(arr[i][1],arr[i][3])>=min(arr[count][1],arr[count][3]))&&(max(arr[count][1],arr[count][3])>=min(arr[i][1],arr[i][3]))){  cout << "Cross again" << endl;  cout << min(arr[i][0],arr[i][2]) << " " << min(arr[i][1],arr[i][3]) << " / " << max(arr[i][0],arr[i][2]) << " " << max(arr[i][1],arr[i][3]) << endl;  int p1\_x = arr[i][0];  int p1\_y = arr[i][1];  int p2\_x = arr[i][2];  int p2\_y = arr[i][3];    int p3\_x = arr[count][0];  int p3\_y = arr[count][1];  int p4\_x = arr[count][2];  int p4\_y = arr[count][3];  if ((((p3\_x-p1\_x)\*(p2\_y-p1\_y)-(p3\_y-p1\_y)\*(p2\_x-p1\_x)) \* ((p4\_x-p1\_x)\*(p2\_y-p1\_y)-(p4\_y-p1\_y)\*(p2\_x-p1\_x)))<=0){  if ((((p1\_x-p3\_x)\*(p4\_y-p3\_y)-(p1\_y-p3\_y)\*(p4\_x-p3\_x)) \*((p2\_x-p3\_x)\*(p4\_y-p3\_y)-(p2\_y-p3\_y)\*(p4\_x-p3\_x)))<=0){  cout << "Lines cross";  break;  }  }    }  }  }  }  arr[count][0]<p1\_min?p1\_min=arr[count][0]:arr[count][0]>p1\_max?p1\_max=arr[count][0]:true;  arr[count][1]<p2\_min?p2\_min=arr[count][1]:arr[count][1]>p2\_max?p2\_max=arr[count][1]:true;  count++;  }  int min(int& a, int& b){ if (a<b) return a; return b;}  int max(int& a, int& b){ if (a>b) return a; return b;}  };  int main() {  int N;  int p1, p2;  // cout << "Enter N: ";  // cin >> N;  N=5;  Broken br(N);  br.put(2,2);  br.put(5,3);  br.put(7,6);  br.put(2,0);  // for (int i =0; i<N; i++){  // cin >> p1 >> p2;  // br.put(p1,p2);  // }  } |

Приклад виконання програми:

|  |
| --- |
| Rectangulars cross  2 0: 0 0 / 5 3  Cross again  0 0 / 2 2  Cross again  2 2 / 5 3  Lines cross  --------------------------------  Process exited after 0.115 seconds with return value 0  Press any key to continue . . . |

Висновок

Я ознайомився з алгоритм перевірки на перетин двох відрізків. Зміст алгоритму полягає правил лінійної алгебри щодо векторного добутку векторів.