原创 【POJ 2653】Pick-up sticks 判线段与线段交

2019-05-04 20:09:09 我是一只计算鸡 阅读数 35 更多

Pick-up sticks

Time Limit: 3000MS		**Memory Limit:** 65536K
Total Submissions: 15600		**Accepted:** 5893

Description

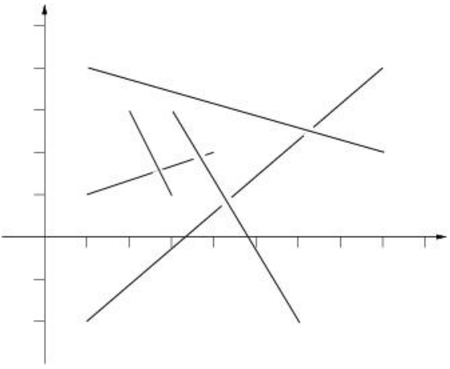
Stan has n sticks of various length. He throws them one at a time on the floor in a random way. After finishing throwing, Stan tries d sticks, that is these sticks such that there is no stick on top of them. Stan has noticed that the last thrown stick is always on top but a all the sticks that are on top. Stan sticks are very, very thin such that their thickness can be neglected.

Input

Input consists of a number of cases. The data for each case start with 1 <= n <= 100000, the number of sticks for this case. The foll g contain four numbers each, these numbers are the planar coordinates of the endpoints of one stick. The sticks are listed in the orde wh thrown them. You may assume that there are no more than 1000 top sticks. The input is ended by the case with n=0. This case should not processed.

Output

For each input case, print one line of output listing the top sticks in the format given in the sample. The top sticks should be listed in orde they were thrown.



The picture to the right below illustrates the first case from input.

Sample Input

```
5
1 1 4 2
2 3 3 1
1 -2.0 8 4
1 4 8 2
3 3 6 -2.0
3
0 0 1 1
1 0 2 1
2 0 3 1
0
```

Sample Output

```
Top sticks: 2, 4, 5.
Top sticks: 1, 2, 3.
```

题意：有顺序的扔一些木棍，让你求最上面的木棍。

题解：如果是最上面的木棍，那么后面就没有木棍和它相交。相当于判断后面的线段是否和当前线段相交。

```
#include<cstdio>
#include<cmath>
#include<cstring>
using namespace std;
const int maxn = 100000+7;
const double eps = 1e-8;
double min(double x, double y)
{
  return x < y ? x : y;
}
double max(double x, double y)
{
  return x > y ? x : y;
}
int sgn(double x)
{
  if(fabs(x) < eps) return 0;
  else if(x < 0) return -1;
  else return 1;
}
struct point{
  double x,y;
  point(){}
  point(double _x, double _y) {
    x = _x;
    y = _y;
  }
  point operator - (const point &b) const{
    return point(x-b.x, y-b.y);
  }
  double operator * (const point &b) const{  //点积
    return x*b.x + y*b.y;
  }
  double operator ^ (const point &b) const{  //叉积
    return x*b.y - y*b.x;
  }
};
struct line{
  point s,e;
  line(){}
  line(point _s, point _e) {
    s = _s;
    e = _e;
  }
};
line l[maxn];
int n;
bool flag[maxn];
bool inter(line l1,line l2)  // 判线段交
{
  return
        max(l1.s.x,l1.e.x) >= min(l2.s.x,l2.e.x)&&//快
        max(l1.s.y,l1.e.y) >= min(l2.s.y,l2.e.y)&&//速
```

展开阅读全文