转载 凸包--Graham扫描法

2019-02-16 11:26:58 _-Y-_-Y-_ 阅读数 38 更多

原文:https://www.cnblogs.com/cjyyb/p/7260523.html

(大佬写的贼好，一看就懂的哪种，膜拜大佬)

例题

Surround the Trees HDU - 1392
https://cn.vjudge.net/problem/HDU-1392
There are a lot of trees in an area. A peasant wants to buy a rope to surround all these trees. So at first he must know the minimal require
the rope. However, he does not know how to calculate it. Can you help him?
The diameter and length of the trees are omitted, which means a tree can be seen as a point. The thickness of the rope is also omitted wh
rope can be seen as a line.

There are no more than 100 trees.
Input
The input contains one or more data sets. At first line of each input data set is number of trees in this data set, it is followed by series of c
of the trees. Each coordinate is a positive integer pair, and each integer is less than 32767. Each pair is separated by blank.

Zero at line for number of trees terminates the input for your program.
Output
The minimal length of the rope. The precision should be 10^-2.
Sample Input
9
12 7
24 9
30 5
41 9
80 7
50 87
22 9
45 1
50 7
0
Sample Output
243.06
(大佬写的很清楚，我就不写什么了，直接上代码)

代码

```
#include <bits/stdc++.h>
using namespace std;
const int INF=0x3f3f3f;
struct Node{
    int x,y;
}p[105],S[105];
int n,top;
inline bool cmp(Node a,Node b)//比较函数，对点的极角进行排序
{
       double A=atan2((a.y-p[1].y),(a.x-p[1].x));
       double B=atan2((b.y-p[1].y),(b.x-p[1].x));
       if(A!=B)return A<B;
       else    return a.x<b.x; //这里注意一下，如果极角相同，优先放x坐标更小的点
}
long long Cross(Node a,Node b,Node c)//计算叉积
{
       return 1LL*(b.x-a.x)*(c.y-a.y)-1LL*(b.y-a.y)*(c.x-a.x);
}
```

```
void Get()//求出凸包
{
       p[0]=(Node){INF,INF};int k;
       for(int i=1;i<=n;++i)//找到最靠近左下的点
              if(p[0].y>p[i].y||(p[0].y==p[i].y&&p[i].x<p[0].x))
               {p[0]=p[i];k=i;}
       swap(p[k],p[1]);
       sort(&p[2],&p[n+1],cmp);//对于剩余点按照极角进行排序
       S[0]=p[1],S[1]=p[2];top=1;//提前在栈中放入节点
       for(int i=3;i<=n;)//枚举其他节点
       {
              if(top&&Cross(S[top-1],p[i],S[top])>=0)
                        top--;//如果当前栈顶不是凸包上的节点则弹出
              else  S[++top]=p[i++];//加入凸包的栈中
       }
}
int main() {
    while(~scanf("%d", &n)&&n){
        memset(p,0,sizeof(p));
        memset(S,0,sizeof(S));
        top=0;
        for(int i=1;i<=n;i++)
            scanf("%d %d", &p[i].x, &p[i].y);
        Get();
        double ans=0.0;
        if(top==0)
        ans=0;
        else if(top==1)
        ans=pow(pow((S[0].x-S[top].x),2)+pow(S[0].y-S[top].y,2),0.5);
        else{
            for(int i=0;i<top;i++){
                ans+=pow(pow((S[i].x-S[i+1].x),2)+pow(S[i].y-S[i+1].y,2),0.5);
            }
            ans+=pow(pow((S[0].x-S[top].x),2)+pow(S[0].y-S[top].y,2),0.5);
        }
        printf("%.2lf\n", ans);
    }
}
```

有 0 个人打赏

文章最后发布于: 20