【题意简述】

给定一个所有边均平行于坐标轴的简单多边形，求它内部最大的圆的半径。

【算法分析】

二分半径，判断时，假设半径为r的圆在多边形内，我们可以移动这个圆，使它满足以下3个条件的其中一个：

1. 贴着两条互相垂直的边。

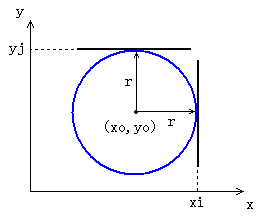
2. 贴着一条边和一个顶点。

3. 贴着两个顶点。

然后求出候选的圆心坐标，然后逐一判断是否合法。

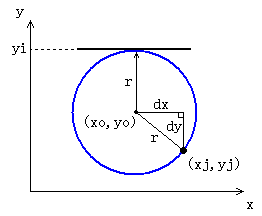
下面对3种情况分别讨论如何求出候选的圆心坐标。

1. 直升机场贴着两条互相垂直的边。



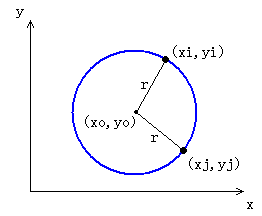
假设与y轴平行的边的x坐标为*xi*，与x轴平行的边的y坐标为*yi*，则候选的圆心坐标有4个：(*xi*+*r*, *yi*+*r*)、(*xi*+*r*, *yi*-*r*)、(*xi*-*r*, *yi*+*r*)和(*xi*-*r*, *yi*-*r*)。

2. 直升机场贴着一条边和一个顶点。



以与x轴平行的边为例，与y轴平行的边可以类似地处理。假设与x轴平行的边的y坐标为*yi*，顶点坐标为(*xj*, *yj*)，则候选的圆心的y坐标为*yo*=*yi*-*r*或*yo*=*yi*+*r*。如果*yj*<*yi*，则取*yo*=*yi*-*r*，否则取*yo*=*yi*+*r*，求得*dy*=|*yo*-*yj*|。如果*dy*>*r*则没有候选的方案，否则可以求得*dx*=sqrt(*r*\**r*-*dy*\**dy*)，候选的圆心坐标为(*xj*+*dx*, *yo*)和(*xj*-*dx*, *yo*)。

3. 直升机场贴着两个顶点。



假设顶点坐标为(*xi*, *yi*)和(*xj*, *yj*)，通过解方程组

(*xi*-*xo*)\*(*xi*-*xo*)+(*yi*-*yo*)\*(*yi*-*yo*)=*r*\**r*

(*xj*-*xo*)\*(*xj*-*xo*)+(*yj*-*yo*)\*(*yj*-*yo*)=*r*\**r*

即可求得圆心坐标(*xo*,*yo*)。

圆在多边形内需要满足两个条件：

1. 圆心在多边形内；

2. 多边形的边不与圆相交。

至此，问题完全解决。

【复杂度分析】

时间复杂度：枚举O(N^2)，判断O(N)，若二分范围为M（包括精度），则总复杂度O(N^3logM)

空间复杂度：O(N)

【注意点】

一定要注意精度！

【程序清单】

#include <cstdio>

#include <cstdlib>

#include <cstring>

#include <cctype>

#include <cmath>

const double eps=1e-6;

int x[25],y[25],n,i,len,px,py,cases;

double ra,rb,r;

char dr;

bool check(double ox,double oy){

int i,s;

s=0;

for(i=0;i<n;i++)

if(x[i]>ox&&((y[i]>oy)^(y[i+1]>oy))) s++;

/\*这种判断方法是向x轴正方向射出一条射线，并不严密，不过所幸没有被卡；其实该题判断点在多边形内可以向y=x方向射线，也可用转角法等\*/

if(s%2==0)return false;

for(i=0;i<n;i++){

if((x[i]-ox)\*(x[i]-ox)+(y[i]-oy)\*(y[i]-oy)<(r-eps)\*(r-eps)) return false;

if(x[i]==x[i+1]&&((y[i]>oy)^(y[i+1]>oy))&&fabs(x[i]-ox)<r-eps) return false;

if(y[i]==y[i+1]&&((x[i]>ox)^(x[i+1]>ox))&&fabs(y[i]-oy)<r-eps) return false;

}

return true;

}

bool ok(){

int i,j;

double di,dd,mx,my,dx,dy;

for(i=0;i<n;i++)

if(x[i]==x[i+1])

for(j=0;j<n;j++)

if(y[j]==y[j+1]){

if(check(x[i]+r,y[j]+r))return true;

if(check(x[i]+r,y[j]-r))return true;

if(check(x[i]-r,y[j]+r))return true;

if(check(x[i]-r,y[j]-r))return true;

}

for(i=0;i<n;i++)

for(j=0;j<n;j++)

if(x[i]==x[i+1]){

di=fabs(x[j]-(x[i]+r));

if(di<r){

dd=sqrt(r\*r-di\*di);

if(check(x[i]+r,y[j]+dd))return true;

if(check(x[i]+r,y[j]-dd))return true;

}

di=fabs(x[j]-(x[i]-r));

if(di<r){

dd=sqrt(r\*r-di\*di);

if(check(x[i]-r,y[j]+dd))return true;

if(check(x[i]-r,y[j]-dd))return true;

}

}

else{

di=fabs(y[j]-(y[i]+r));

if(di<r){

dd=sqrt(r\*r-di\*di);

if(check(x[j]+dd,y[i]+r))return true;

if(check(x[j]-dd,y[i]+r))return true;

}

di=fabs(y[j]-(y[i]-r));

if(di<r){

dd=sqrt(r\*r-di\*di);

if(check(x[j]+dd,y[i]-r))return true;

if(check(x[j]-dd,y[i]-r))return true;

}

}

for(i=0;i<n-1;i++)

for(j=i+1;j<n;j++){

mx=(x[i]+x[j])/2.0;

my=(y[i]+y[j])/2.0;

di=sqrt((x[i]-mx)\*(x[i]-mx)+(y[i]-my)\*(y[i]-my));

if(di>0&&di<r){

dd=sqrt(r\*r-di\*di);

dx=(my-y[i])/di\*dd;

dy=(x[i]-mx)/di\*dd;

if(check(mx+dx,my+dy))return true;

if(check(mx-dx,my-dy))return true;

}

}

return false;

}

int main(){

freopen("b.in","r",stdin);

freopen("b.out","w",stdout);

while(scanf("%d",&n)&&n){

if(cases)printf("\n");

px=0;py=0;

for(i=1;i<=n;i++){

scanf("%d %c",&len,&dr);

if(dr=='R')px+=len;

if(dr=='L')px-=len;

if(dr=='U')py+=len;

if(dr=='D')py-=len;

x[i]=px;

y[i]=py;

}

ra=0;rb=999;

while(rb-ra>eps){

r=(ra+rb)/2;

if(ok())ra=r;

else rb=r;

}

printf("Case Number %d radius is: %.2lf\n",++cases,r);

}

return 0;

}

赵亮