

# 梯形图(trapezoidal map)实验

费雨果 2009400420

苟丽美 D0916018

徐博林 2009210951

## 一、 实验目的

1. 通过实验，阅读了解点定位问题的资料，完成对梯形图的初步了解；
2. 通过实验，实现梯形图结构的构造及查询算法，并通过对它的性能分析，了解此算法的优劣；
3. 通过实验锻炼相互交流能力，对以后的科研工作打好基础

## 二、 实验环境

WinXP Visual Studio 2008

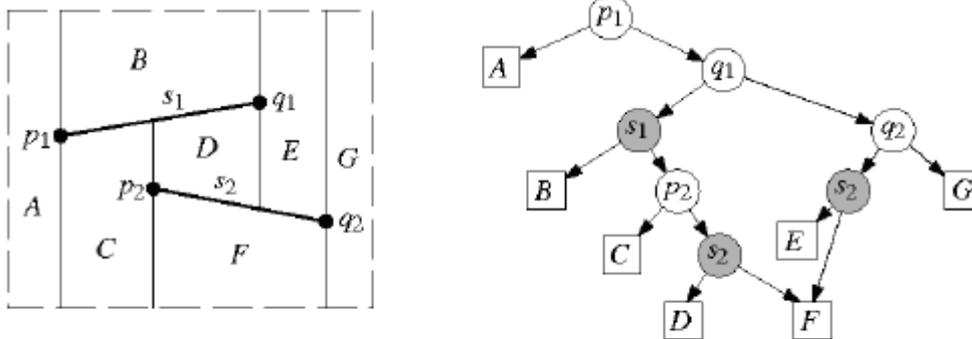
编译 C#

## 三、 实验分析

按照课件所讲，梯形图结构的构造与查询是通过建立一个随机增量式算法 (randomized incremental algorithm), 利用它为任意一组线段集  $S$ , 构造出对应的梯形图  $T(S)$ , 在构造梯形图过程中, 同时构造一个数据结构  $D$ , 借助它在  $T(S)$  中进行点定位查询。

该算法所构造的数据结构  $D$  是衣服有向无环图的查找结构, 其中有唯一的根节点, 同时对应于  $S$  的梯形图中的每个梯形, 有且仅有一个叶子。每个内部节点的度为 2。所有内部节点分为两类:  $x$ -节点和  $y$ -节点。每个  $x$ -节点用  $S$  中某条线段的一个端点作为判断准则; 每个  $y$ -节点用  $S$  中某条线段作为判断准则。

例如下图:



两条线段的梯形图, 及其对应的查找结构

我们在对点  $p$  进行查询时, 从根节点出发, 沿着某条有向路径, 每遇到一个新节点都将其与  $p$  比较, 若是  $x$ -节点, 取该点横坐标与  $p$  横坐标比较, 若  $p$  较大则选右子树, 否则选

左子树，若是  $y$ -节点，取  $p$  投影到节点所存线段的点与  $p$  比较， $p$  在下面选左子树，否则选右子树。最终到达的那个叶子就对应于  $T(S)$  中包含  $p$  的那个梯形。

#### 四、 实验算法

我们按照课件上给出的构造查找结构的递增式算法

Algorithm ConstructTrapezoidalMap(S)

1、构造一个包围框  $R$  包含  $S$  中所有线段;

2、 $TM(S)$  = 单个梯形图  $R$ ,

$SS(S)$  = 单个表示  $R$  的根节点;

3、对每条属于  $S$  的线段  $s=pq$  //可先将  $S$  中线段顺序打乱得到一个随机序列  
找到  $p$  所在的梯形令其为  $Trap$ ，做循环操作

(1) 对梯形  $Trap$  进行替换操作，并更新  $TM(S)$ 和  $SS(S)$ ;

(2) 比较  $Trap$  右端点  $rightp$  与线段  $s$  的位置，若  $rightp$  在  $s$  的上方

$Trap=Trap$  的右下方梯形，

否则  $Trap=Trap$  的右上方梯形

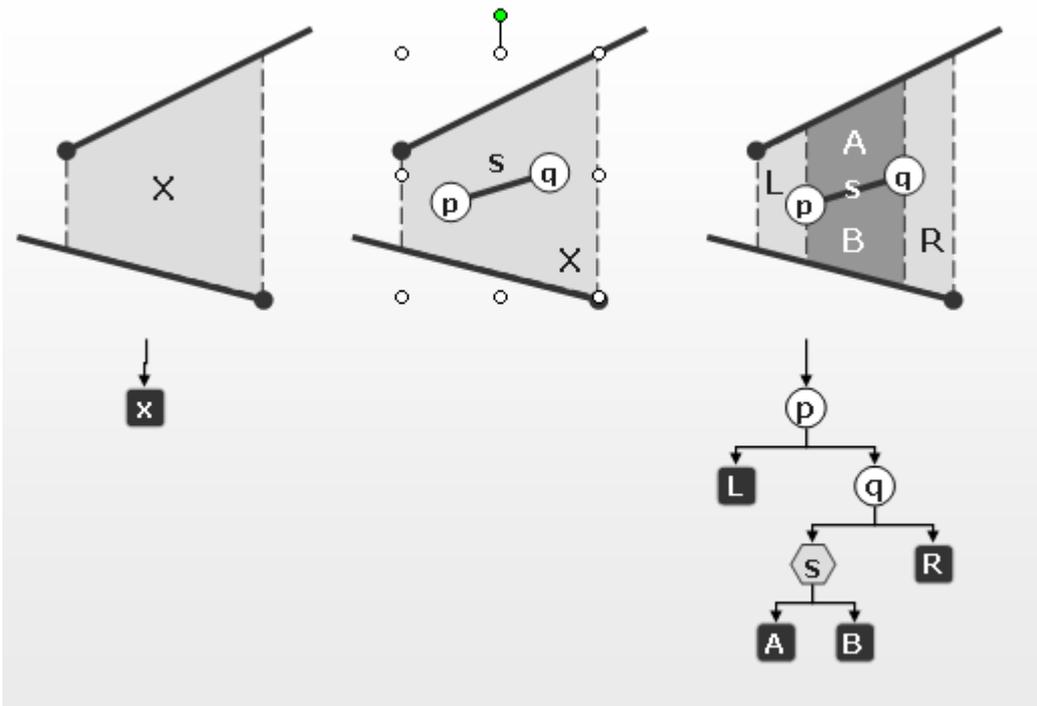
直至  $q < Trap \rightarrow rightp$  为空。

替换操作

总共有三种情况:

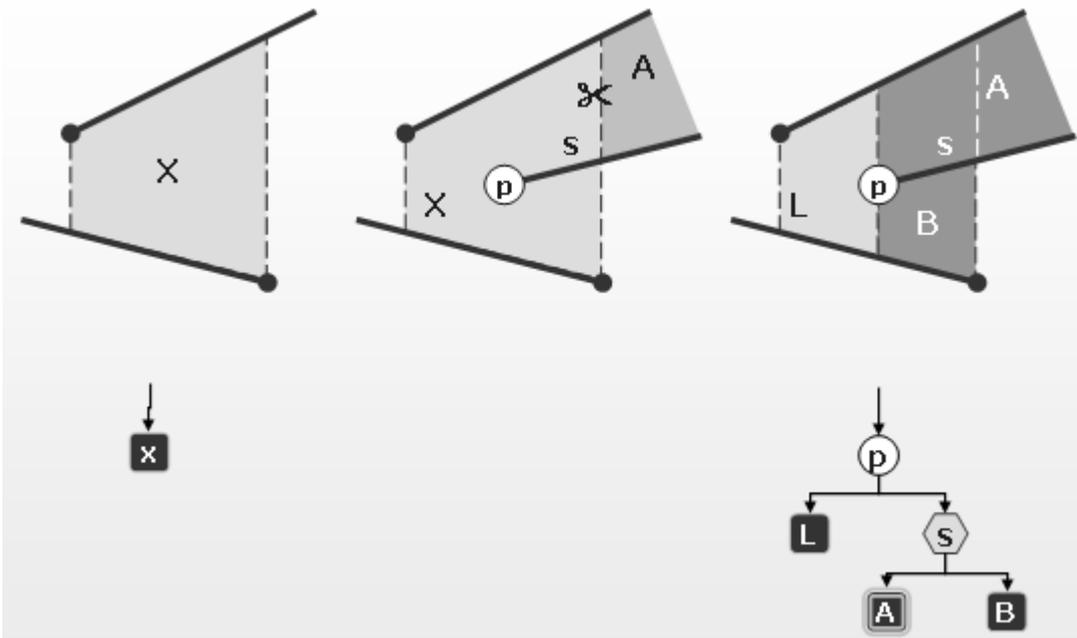
1. 两端点均在  $Trap$  中

这种情况最容易做



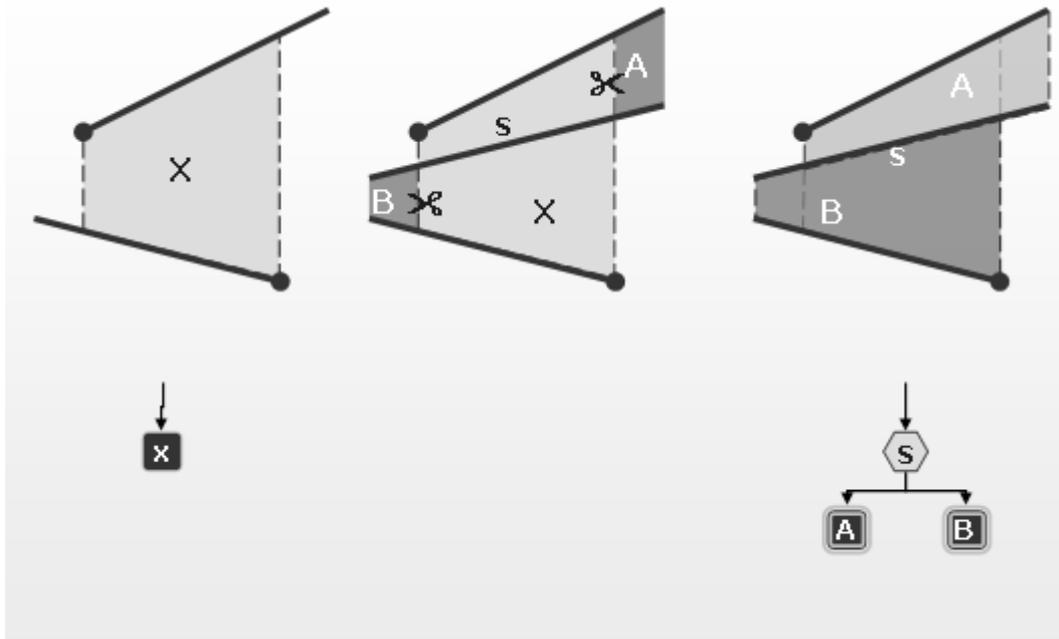
修改 TM (S) 中 DCEL 结构，并替换 SS (S) 中叶节点 X

2. 一个端点在 Trap 中  
不妨，令那个端点是 p



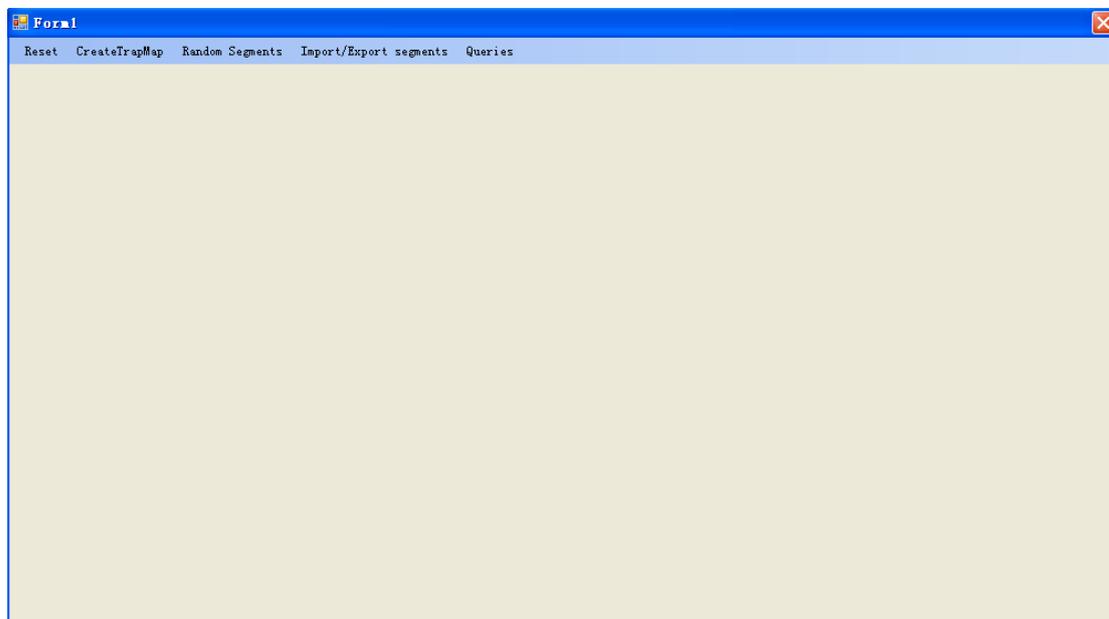
其中判断哪个叶节点是有两个指向的需要比较 s 与 Trap->rightp 位置

3. 无端点在 Trap 中

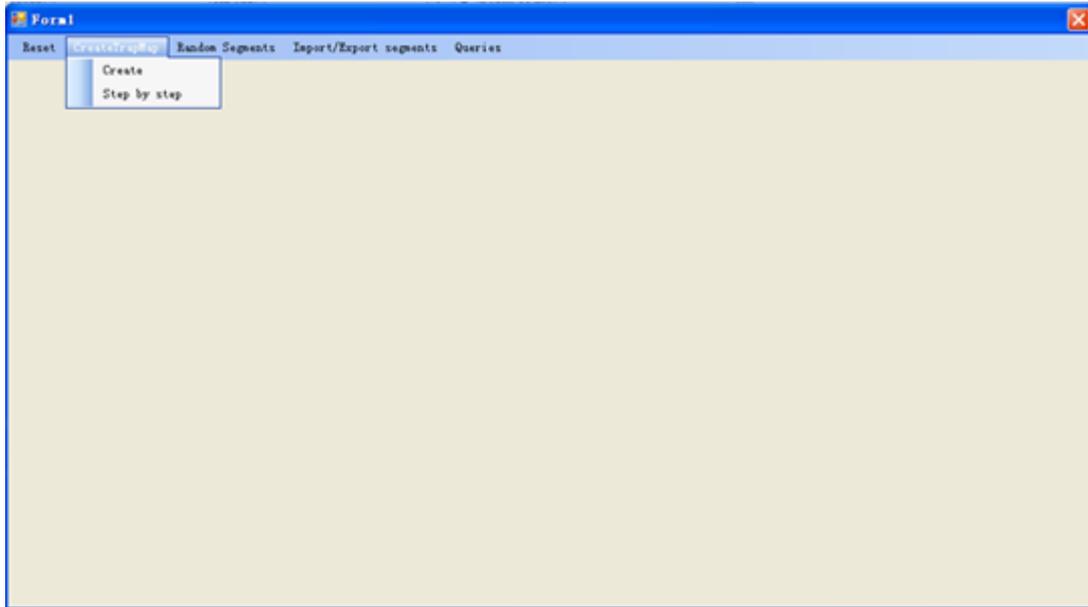


## 五、 实验结果与分析

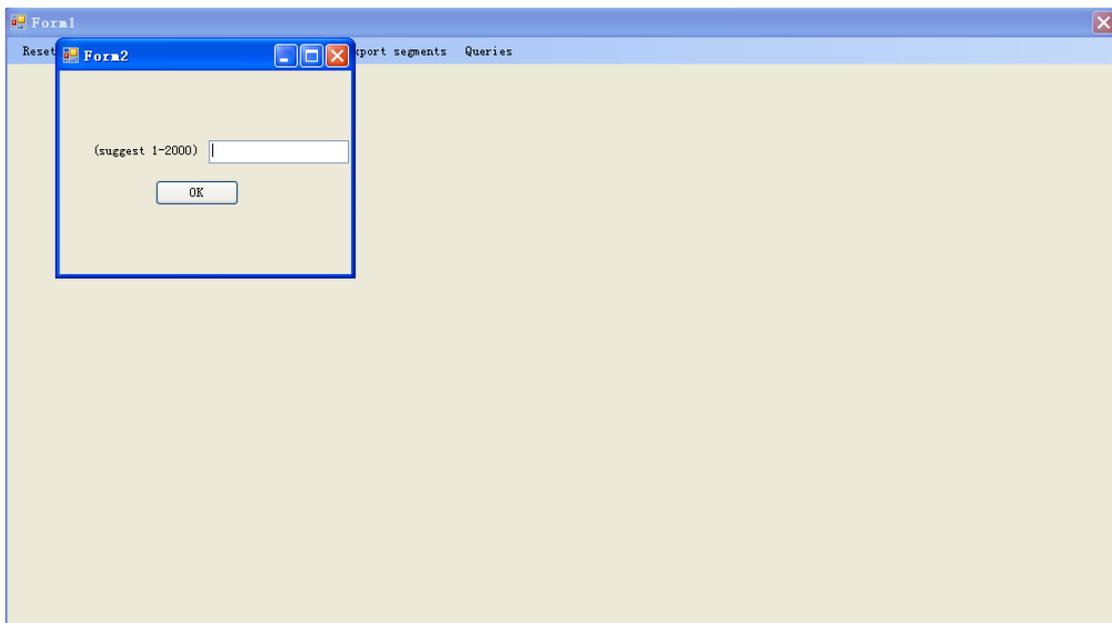
运行程序得到如下界面



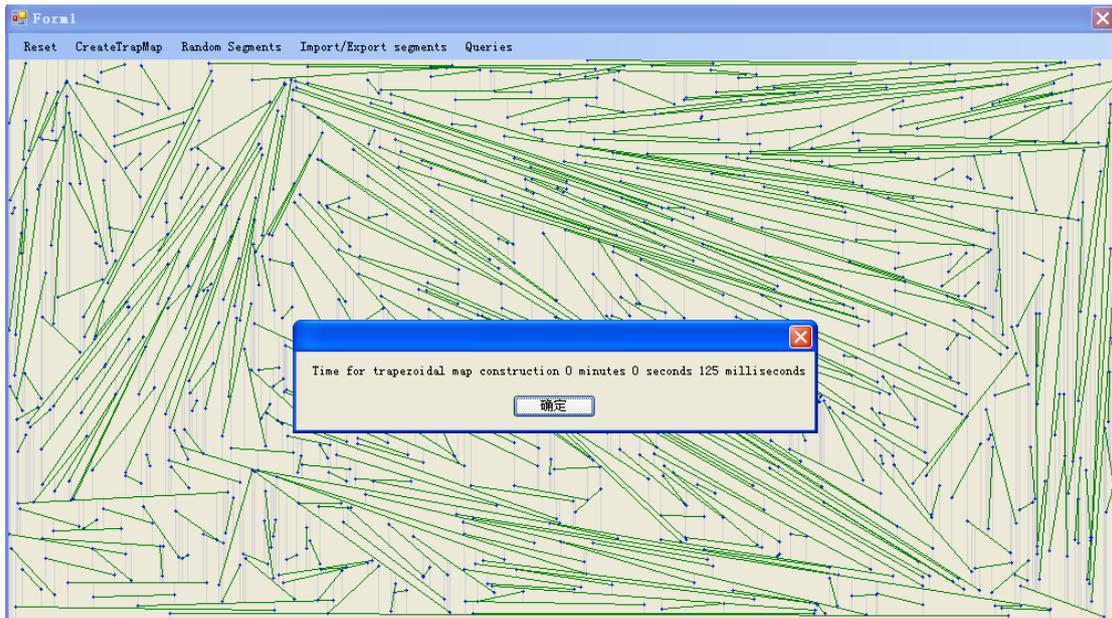
菜单第一个实现清空操作



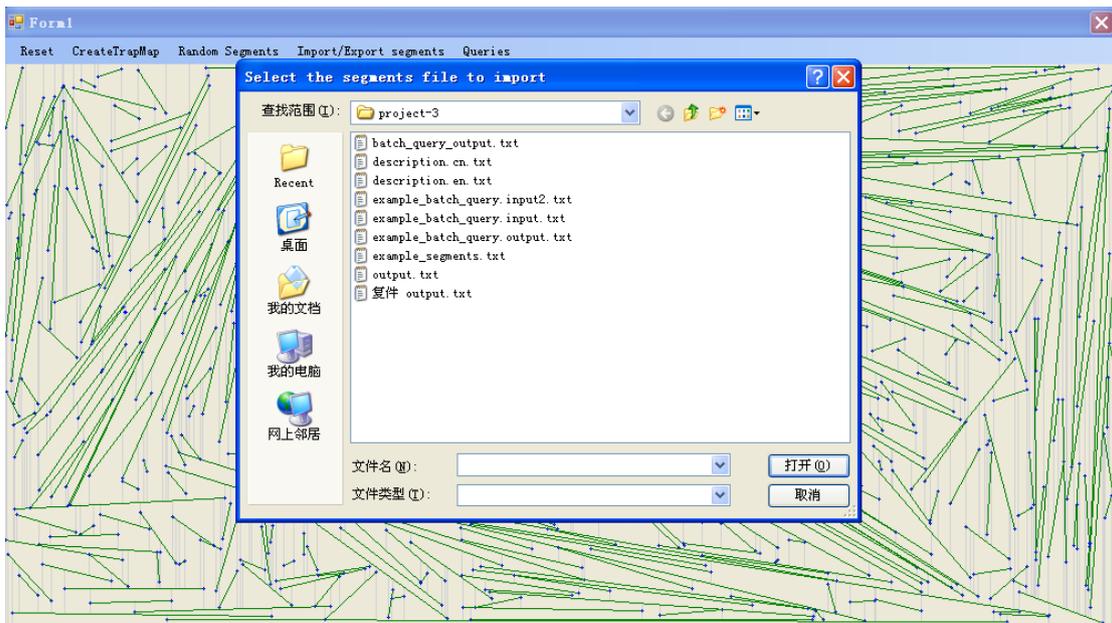
菜单第二个按钮实现 TrapMap 的建立和单步建立（这里推荐 20 以下的）

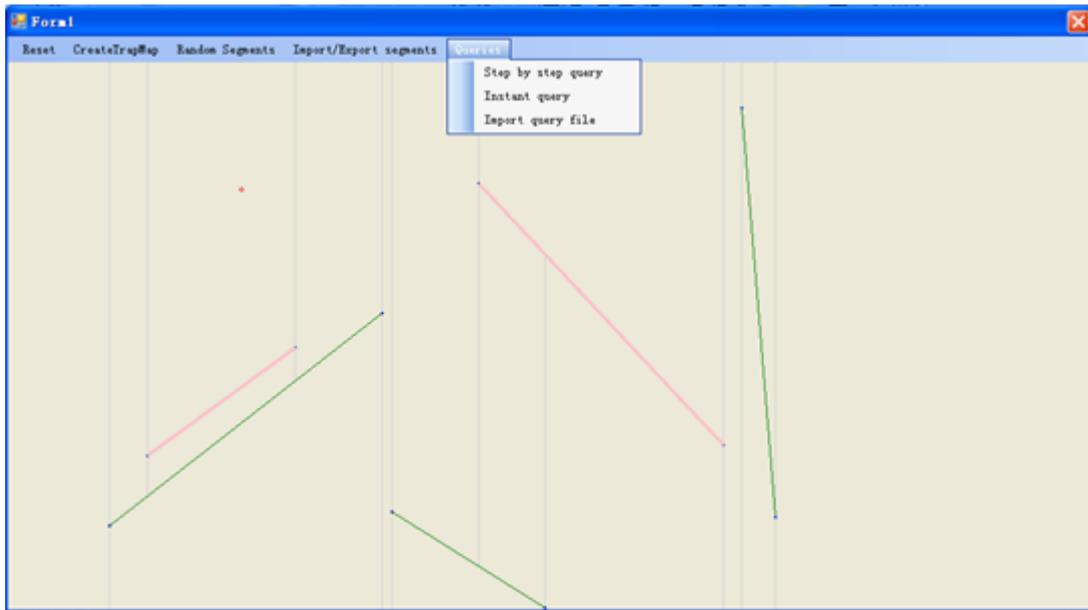


菜单第三个用来实现不交叉线段的生成，因为采用算法比较简略，没做任何优化，建议 2000 以下，过多的话，程序可能崩溃  
生成线段后调用菜单第二个 Creat，就可生成 Trapmap



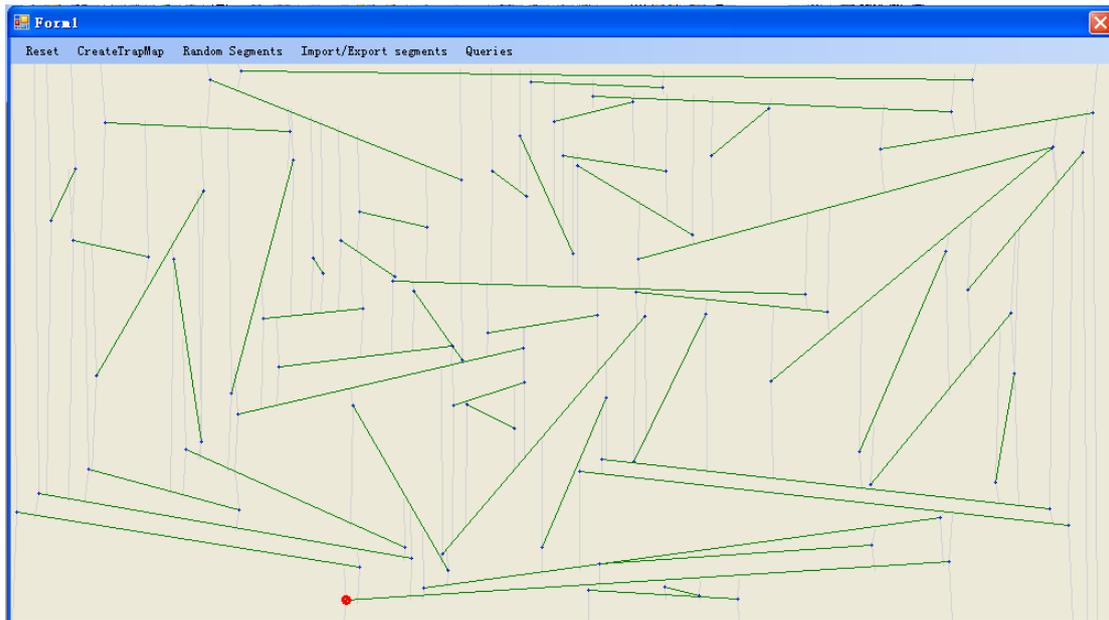
#### 第四个用来输入输出线段集





第五个菜单栏用来查找点，其中第一个是动态演示，第二个是瞬间查询，第三个是批量导入查询

动态演示如下图：



红色端点表明正在搜寻比较的线段，搜到最后结果后显示点所属梯形的下线段



对多组查询统计表格如下：

线段数	构造时间 (ms)	1 万个点查询时间 (ms)	10 万个点查询时间 (ms)
50	-	-	79
100	16	-	78
500	47	16	125
1000	110	32	156
5000	484	47	265
10000	1437	78	313

-表明时间几乎已经测不出来

对上述数据分析可以发现构造梯形图所画时间基本在  $O(n \log n)$  对点的点定位查找，基本可以在  $O(\log n)$  的时间内完成

## 六、 实验收获

通过此次实验我们对点定位，特别是采取随机增量式算法构造梯形图有了一个比较深刻的理解。由于此次实验时间比较仓促，特别为了准备考试，我们缺乏足够沟通，以至于很多东西没有实现，十分遗憾。