ACM/ICPC World Finals 2013 G

Map Tiles 解题报告

合肥一中 梁泽宇

【题目大意】

给出一个简单多边形，用若干指定长宽的矩形来覆盖它，要求矩形的边平行于坐标轴，不能旋转，且两个相邻的矩形只能在整条边上接触，求至少需要多少个矩形才能将这个多边形完全覆盖。

数据范围：3<=多边形的边数 n<=50，设矩形的长、宽分别为 xs 和 ys，则多边形所有顶点的坐标范围满足 $0<=x<=10xs$，$0<=y<=10ys$（xs 和 ys 的范围对于解决本题无意义）。

【考察算法】

计算几何（平移思想、扫描线思想、线段与多边形相交等）

【分析】

首先注意到，本题对矩形放置位置的要求使得，只要确定了任意一个矩形的位置，其它矩形的位置也就唯一确定，一个方案也就唯一确定了。因此本题的算法分为两步，第一步，确定一个矩形的位置；第二步，计算由这个矩形确定的方案需要多少个矩形。

先来看第一步。由于本题的矩形可以放在任意实数位置，所以方案总数是无限的。但是，我们可以通过“平移”思想将方案个数变为有限。考虑一种方案，在确定了需要哪些矩形之后，多边形可以在这些矩形形成的“框架”中任意平移，直到出现以下三种情况之一为止：

（1）多边形上有两个（或两个以上）的顶点碰到了框架的边界，且其中一个碰到的是水平边界，另一个碰到的是竖直边界（注意，这两个顶点必须是不同的，如果是同一个顶点同时碰到水平边界与竖直边界，即与框架的一个顶点重合，则它还可以继续平移）；

（2）多边形上有一个（或以上的）顶点碰到了框架的边界（水平或竖直），且有一条（或以上的）边碰到了框架的顶点。注意，在这种情况中，如果多边形的顶点碰到的是水平边界，则那条碰到框架顶点的边不能是水平的，否则它还可以继续平移，竖直的同理；

（3）多边形上有两条（或以上的）边碰到了框架的顶点，且这两条边不平行（如果平行，则多边形还可以沿着这两条边的方向继续平移）；

对于第（1）种情况，只需要枚举这两个顶点（设碰到水平边界的顶点为 i、碰到竖直边界的顶点为 j），就可以直接得到其中一个矩形顶点的坐标为（j.x, i.y)。

对于第（2）种情况，显然要枚举多边形中碰到的框架边界的顶点 i，和碰到了框架顶点的边 e，但是，仅仅枚举这两个量还是不够的，因为边 e 上可能有多个点，满足它们到 i 所碰到的边界的距离为 *xs*（竖直边界）或 *ys*（水平边界）的倍数，此时这些点也都要枚举，才能确定一个矩形的位置；

对于第（3）种情况，要枚举这两条边，同时要枚举它们碰到的框架的顶点的 X、Y 坐标之差分别是 *xs*、*ys* 的几倍，然后可以列出一个二元一次方程组，由于这两条边不平行，所以它有唯一解，解出后，先要判断一下这个解是否在该边（线段）上，如果在该边上，才能确定一种方案。

下面计算第一步的枚举量。注意（2）（3）当中需要枚举的倍数都是在-10～10 范围内的（因为题目限制了多边形的坐标范围不超过 *xs*、*ys* 的 10 倍），也就是说，（1）（2）（3）的枚举量最坏情况下分别为 $O(n^2)$ 、$O(n^2m)$ 、$O(n^2m^2)$，其中 *m*=21。第（3）种情况的枚举量最大，当 n=50 时，枚举量为 50*50*21*21=1102500（其实达不到这么多，因为线段长度是有限的，真正满足解出的两个点都在相应边上的情况不多）。

然后，本题的关键是第二步，即确定一种方案需要多少个矩形，如何实现。

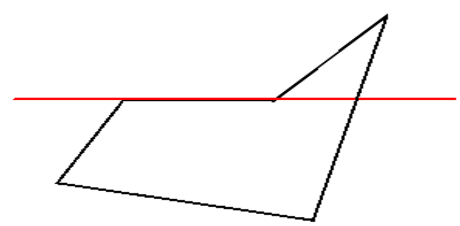
很显然的想法是先将所有坐标范围不超过多边形坐标范围的矩形都列举出来，然后一一判断它们是否需要（如果一个矩形中的任意一个点在多边形内部，则它需要使用，注意，如果它只有边界上的点在多边形的边界上，则不需要使用），由于多边形坐标范围的限制，这些矩形的数量不会很多，最多有 11×11=121 个（之所以是 11 而不是 10，是因为可能矩形并不从 0 开始放），但是，如果暴力实现，一一判断，对每个矩形都要花费 $O(n)$ 的时间，最后整个第二步的时间复杂度会达到 $O(nm^2)$，会 TLE。为了高效解决这个问题，需要引入扫描线的思想。

首先，一个矩形需要使用，当且仅当它的任意一条边和多边形内交（线段和多边形内交即线段上至少有一个点在多边形内部，不含边界，注意，它包括了线段被多边形完全包含的情况）。也就是只需要判断出每个矩形的每条边是否与多边形内交即可。而这些矩形刚好组成一个完整的网格。这样，先求出矩形网格的每条完整线段（水平和竖直方向的所有极大线段）和多边形的各个内交区间（注意是内交区间，也就是与边界重合的部分、一个单点均不计入），然后判断这些线段的每一段是否在这些区间里面即可。

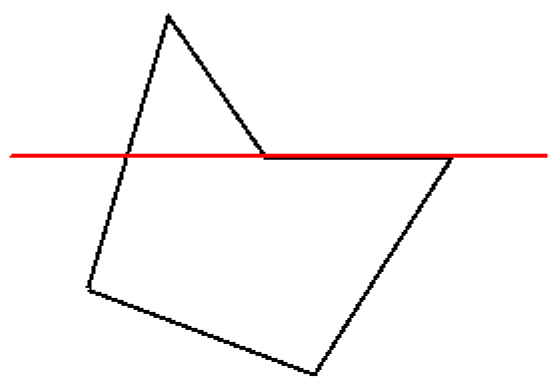
问题是这些内交区间该怎么求？如果暴力求，第二步总的时间复杂度仍然会上升到 $O(nm^2)$，且实现起来非常麻烦。正确方法是，先求出该线段和多边形的所有交点（除了下面所说的与边重合的情形外，与顶点相交的，如果该顶点的两相邻顶点在线段异侧，则计入，否则不计），然后将这些交点排序，则每两个相邻的点组成一个内交区间（容易证明在不出现与边重合的情况下，交点个数必然是偶数）。然而，上述做法忘了考虑一种很重要的情况，就是如果这条线段与多边形的某条边重合，则这条边的两个端点到底是计入交点，还是不计入交点？

具体分析一下这个问题，可以发现，这条边的两个端点是否计入交点，由该边的两个相邻顶点在线段同侧还是异侧，以及该边的左边（水平）或上面（竖直）是否位于多边形内部决定，有四种情况（以下假设该线段是水平的，该线段是竖直的情况同理）：

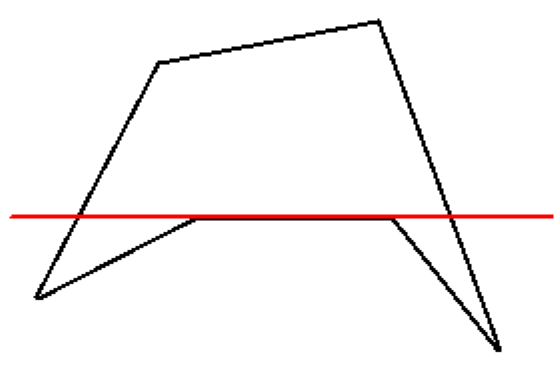
<1>该边的两个相邻顶点在异侧，且该边的左边位于多边形外部，此时该边的左端点不计入而右端点计入：



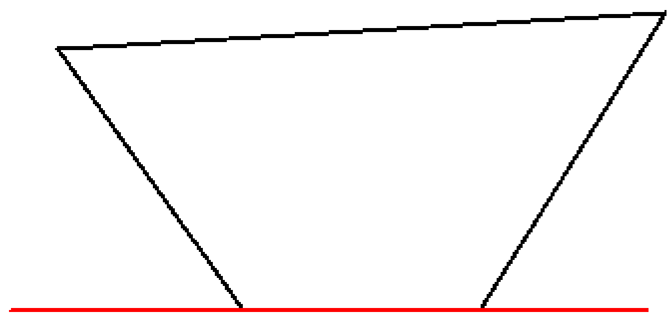
<2>该边的两个相邻顶点在异侧，且该边的左边位于多边形内部，此时该边的右端点不计入而左端点计入：



<3>该边的两个相邻顶点在同侧，且该边的左边位于多边形内部，此时该边的两个端点都要计入，其中左端点作为一个区间的结束，右端点作为另一个区间的开始：



<4> 该边的两个相邻顶点在同侧，且该边的左边位于多边形外部，此时该边的两个端点都不计入：



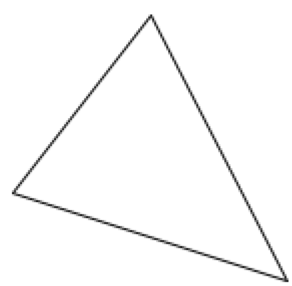
在处理完了这些情况后，第二步就结束了，这种用扫描线思想的方法，时间复杂度仅为 $O(nm)$（m 条线段，多边形 n 条边），配合可行性剪枝可以通过本题了。

【数据说明】

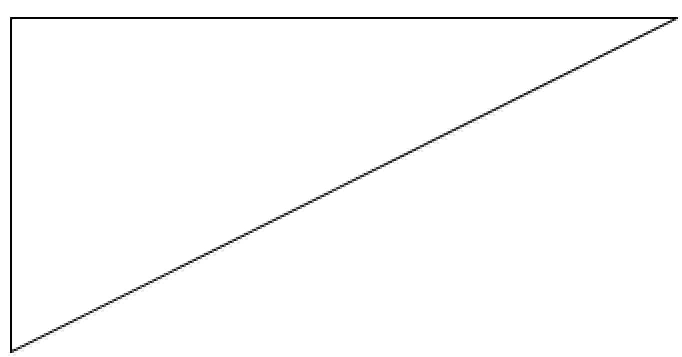
下面是我给本题设计的 10 个测试点（为前 10 个测试点，后 10 个测试点为随机数据）：

（测试点 1～3 为送分）

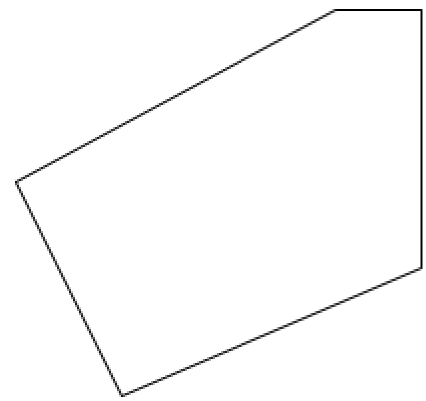
<1>一个三角形，它可以用一个矩形完全覆盖，故结果为 1。设置这个数据主要是为了卡掉忘了判断被一个矩形覆盖的情况的程序（其实只要过了 Sample2 就行了……）：



<2>一个直角边平行于坐标轴的直角三角形：

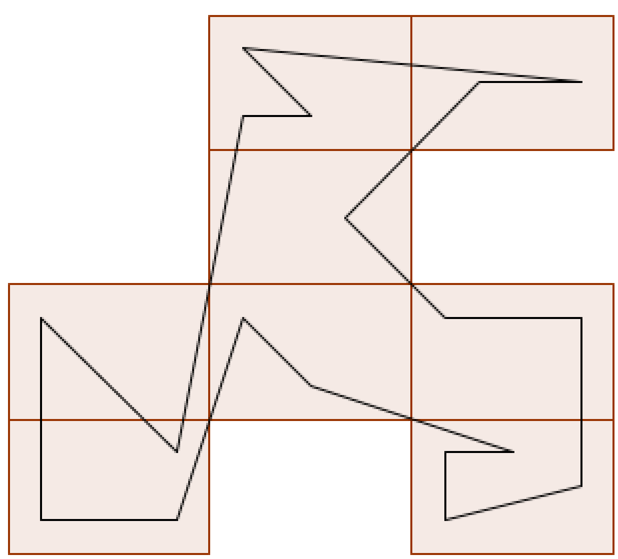


<3>一个凸五边形，其中包含一条水平边和一条竖直边：

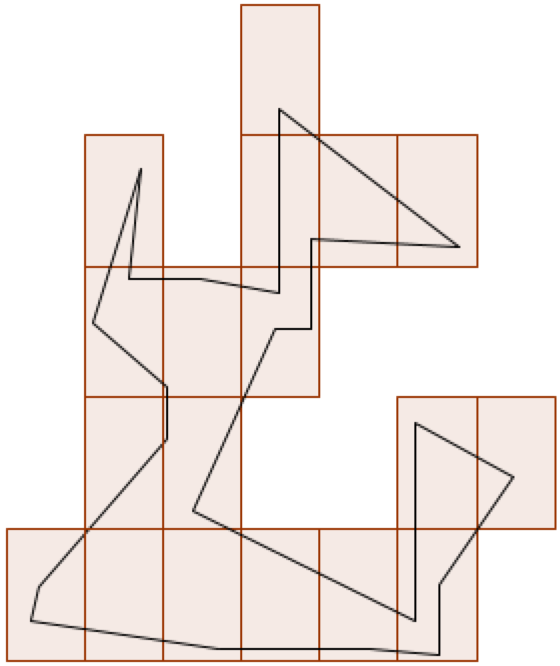


（测试点 4～6 为综合了各种特殊情况的数据，可以卡掉忘了某些细节的程序）

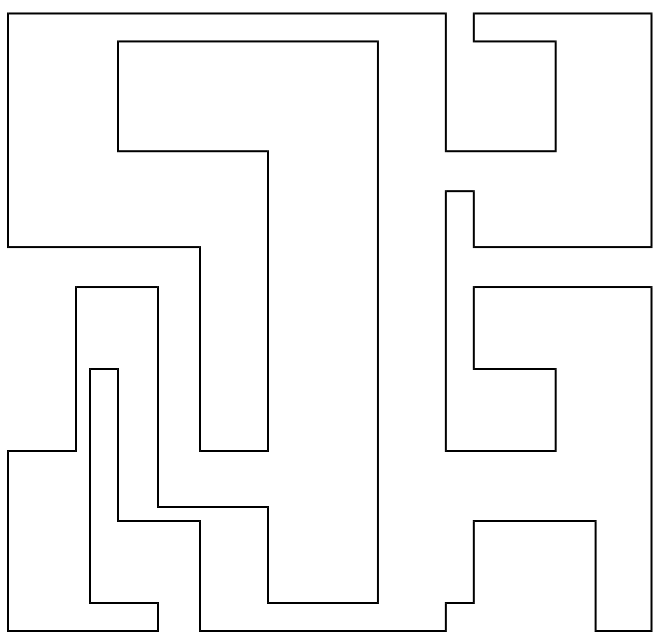
<4>一个含有 18 个顶点的简单多边形，需要 8 个矩形才能完全覆盖，最优方案中，多边形有 4 条边需要碰到框架边界：



<5>一个含有 22 个顶点的简单多边形，需要 18 个矩形覆盖，情况较为复杂：

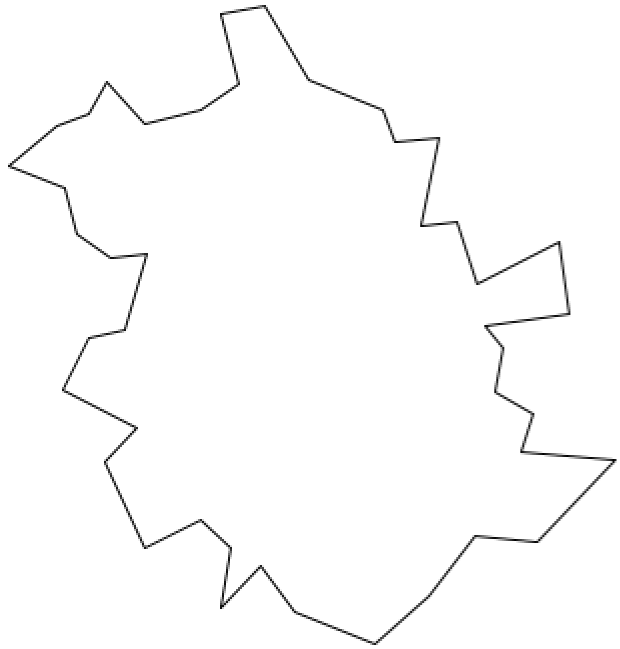


<6>一个 50 个顶点的多边形，其中的所有边均为水平或竖直，设计这个数据主要是为了卡掉忘了判断水平或竖直边的特殊情况的程序：

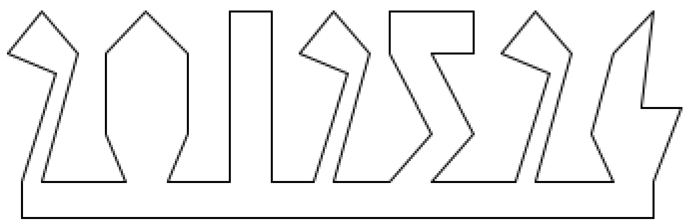


（测试点 7～9 为个性化的原创数据）

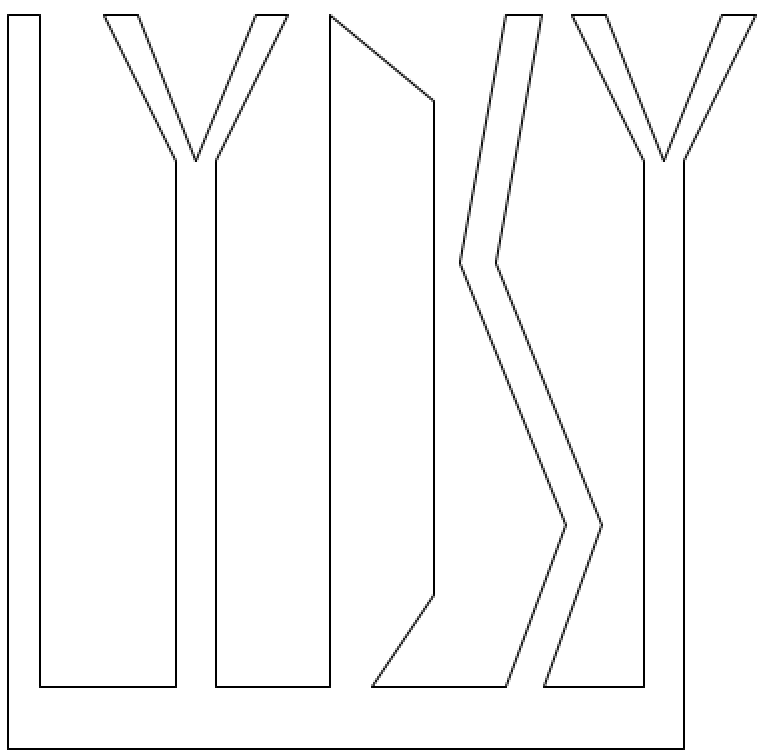
<7>近似的安徽省地图（这题本来就是画地图的……因此弄出个地图）



<8>是一个字符串“2012526”，即 AHOI2012 的日期——纪念 AHOI2012。



<9>是一个字符串”LYDSY”。



（最后是最关键的测试点了）

<10>锯齿形的数据，用来增加第一步中情况（3）的枚举量，从而来卡常数。

本题的官方数据的最后 3 个点都是这种锯齿形的数据。为什么锯齿形的数据可以卡常数？因为，它可以保证其中的每条边的长度都尽可能长，从而增加“倍数”的枚举量。