三角板

【题意简述】

有两种等腰三角形：底边和高相等或底边为高的两倍。依次插入 n 个三角形，将其顶点置于(xi,yi)，底边对齐 x 轴，插入每个三角形后询问目前看得到几个三角形的顶点(不被其他三角形遮住)。若是在插入第 i 个三角形之前(xi,yi)已经位于某三角形内部，则不插入该三角形。

【数据范围】

30%的数据满足：n≤1000
70%的数据满足：n≤40000
100%的数据满足：1≤n≤100000、-10^9≤xi≤10^9、1≤yi≤10^9

【算法分析】

算法 1：

每次插入时，枚举现在所有的三角形，查看它们的顶点是否被其它三角形覆盖。

判断一个点是否在三角形内即判断它是否在直线下方。判断点(x,y)是否在以(xi,yi)为顶点的宽三角形内，即判断是否满足不等式 xi-yi≤x≤xi 且 x-y≥xi-yi(位于左半边)，或满足 xi≤x≤xi+yi 且 x+y≤xi+yi(位于右半边)。判断点(x,y)是否在以(xi,yi)为顶点的窄三角形内，即判断是否满足不等式 2xi-yi≤2x≤2xi 且 2x-y≥2xi-yi(位于左半边)，或满足 2xi≤2x≤2xi+yi 且 2x+y≤2xi+yi(位于右半边)。

时间复杂度: $O(n^3)$

空间复杂度: $O(n)$

期望得分： 15 分

算法 2：

每次插入时，只需检测插入的三角形的顶点是否被覆盖，以及插入前看得到的三角形顶点是否被插入的三角形覆盖即可。

时间复杂度: $O(n^2)$

空间复杂度: $O(n)$

期望得分： 30 分

算法 3：

我们可以用等腰三角形的高将其分为两部分，以宽的三角形的左半边为例。设其顶点为(xi,yi)，则位于其内部的点(x,y)满足 xi-yi≤x≤xi、x-y≥xi-yi。我们可以设计一个高级数据结构：线段树维护 x 在[l,r]之间的所有顶点，并套一棵平衡树维护 x-y 的大小。删除时在相应区间查询即可。由于每个点最多只会删除一次，我们可以在总共 $O(n\log^2 n)$的时间内删除所有被包含的点。

至于如何判断当前插入的三角形顶点是否在某个宽的三角形的左半边内，可以查询 x≥xi，x-y≤xi 的三角形中 x-y 最小的一个，若是 xi-yi≥x-y，则第 i 个三角形的顶点已经被占据。这同样可以用树套树维护。

宽的三角形的右半边、窄的三角形的左、右半边与之类似，这里不再赘述。

时间复杂度:*$O(n\log^2 n)$*

空间复杂度:*$O(n\log n)$*

期望得分：70 分

算法 4：

对于 n 达到 100000 的数据，**算法** 3 依旧难以通过。是不是可以只用线段树或平衡树就能解决此题呢？

仍以宽的三角形的左半边为例。观察不等式 xi-yi≤x≤xi、x-y≥xi-yi，由于 y≥1，可以将其简化为 x≤xi、x-y≥xi-yi。

我们还是用一棵线段树维护 x 坐标。删除在以(xi,yi)为顶点的宽的三角形的左半边内的点，只需查询[-∞,xi]中 x-y 的最大值，如果 x-y≥xi-yi，则将其删除，并更新最大值，继续查询；否则说明该区域内不再有点。

而判断当前插入的三角形顶点是否在某个宽的三角形的左半边内则更加简单，只需查询[xi,∞]中 x-y 的最小值即可。离散化后用树状数组维护，可以做到较低的编程复杂度和常数。

时间复杂度:*$O(n\log n)$*

空间复杂度:*$O(n)$*

期望得分：100 分