

# Solution

---

## fight

---

这个问题本质上是求某条直线的某一侧的点的点权和。

我们可以考虑将所有点放入kdtree中，每个节点除了维护box之外还维护权值和，每次查询的时候用box剪枝。

参见：bzoj 2850

## surround

---

特判只需要 $\leq 2$ 个点的情况。

其余情况需要三个及以上的点，我们for所有的我方点组成的点对，如果敌方点全在我方两个点 $u \rightarrow v$ 的左侧，那么我们连一条边 $u \rightarrow v$ ，这样，问题就等价成了求一个图中的最小环，这个可以用floyd做。

参见：bzoj 1027

## count

---

给定一个简单无向有权图，求其最小生成树的个数。

在我们用Kruskal计算最小生成树时，由于相同权值的边选择的顺序是随机的，所以我们最小生成树就也许有很多。

对于同一权值的边，我们不论用什么顺序“扫过”，最终的得到的无向森林的连通性一定是一样的，即对后面的边是否加入的影响也是一样的，所以可以根据这一点将最小生成树分阶段统计，所有权值相同的边为一阶段，每个阶段都有一个方案数，最终的答案便是方案数的乘积。

对于某一阶段的一个选边的合法方案是什么呢？就是这些边加入到图中，使图的连通性和“按任意顺序扫一遍，能加就加”后的连通性一样。

至于怎么计算，先扫一遍，将减少的连通块的数量记下来，然后撤销操作，枚举边集（ $2^{10}$ ），判断该边集加入后减少的联通块是否一样，一样就合法。

也可以将到达这一阶段时，图中的联通块缩成一个点，然后计算当前阶段的边和缩了点后的图的联通快，每个联通块计算生成树个数，它们的乘积就是本阶段的方案数。

标程用的第二种方法。

参见：bzoj 1016