Problem 16~20

2021年10月17日　　21:57

「2021-10-08 提高模拟赛」火花（sparkle）

先讲一个比正解还难写的 $O(n\log n)$ 做法。对于这个二维平面上的矩形，我们二分一个右边界上的点，用皮克定理（$S=n+\frac{s}{2}+1$）计算出其内部的点的个数。然后我们要找到一个三角形内的斜率指定斜率排名的点，直接 $\mathrm{kth-element}$。

首先这个二分十分鸡肋。我们直接在 sternBrocotTree 上有理数二分；而有理数二分有一个重要结论：若将二分的过程记录下来，写作 LLRRRRLLRLL 的形式，那么连续段只有 $\log n$ 段。所以可以二分跳多少步。

原本而皮克公式已经不适用了，如何计算三角形内部的整点个数呢？设当前二分到的值为 $k=\frac{b}{a}$，则：

$$\sum_{x=1}^{n}\sum_{y=1}^{n}\left[\frac{y}{x}\le k\right]=\sum_{x=1}^{n}\min(n,xk)$$

我们不考虑这个 `min` ，直接考虑后半部分：$\sum_{x=1}^{n}\left\lfloor\frac{xb}{a}\right\rfloor$，直接类欧几里得法计算即可。

「2021-10-09 提高模拟赛」路径统计（path）

对于每种颜色分开统计，统计每个颜色对多少对点有贡献；然后转化成对多少对点没有贡献，显然就是 $\sum_{i=1}^{n}\binom{sz_i}{2}$，其中 sz 为由这个颜色分隔成的每个连通块的大小。

最自然的想法当然是虚树。事实上，直接 dfs 就能做到线性——深搜的过程中，维护 n 个栈，用于维护每个颜色的当前点的最年轻的祖先。再对每个点记录一个 sz，代表它所处的连通块的大小。就显然能做了，搜到一个点时直接把祖先的 sz 剪掉一些就好了嘛。

「2021-10-09 提高模拟赛」两个区间（interval）

单个等差序列的版本，我们相当于要数 $\mathrm{max-min-r+l=0}$ 的区间个数，直接右推右端点维护一个线段树即可了。

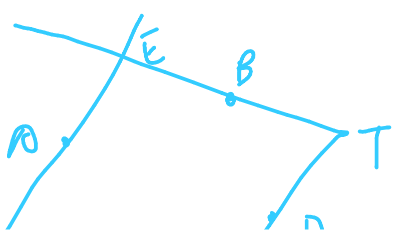
但是这个题中，这种方式已经不适应了，我们直接维护：

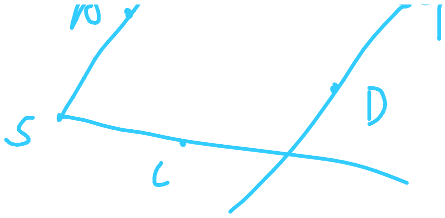
$$\sum_{i=l}^{r}\sum_{j=l}^{r}[a_j=a_i-1]=r-l-1$$

「2021-10-11 提高模拟赛」雾散云开（fish）

考虑四个象限都有点的部分分，我们直接按极角排序，依次经过即可。

考虑有点对跨过超过 π 怎么办？我们再看看从 T 出发是否也有超过 π。如果也有怎么办？如下：



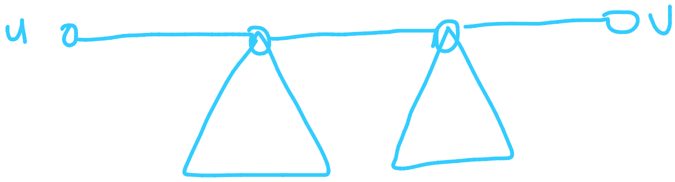


显然对于这个 E 点，我们直接将其他点以它为参考系极角排序，依次输出即可。

[「2021-10-13 提高模拟赛」塔玛丽 (georgian)]

1000^3 居然能过，就说这些。。。。。。

我们分开讨论每对点的贡献，把这条链拉出来看看：



我们发现一些性质：如果出发点在两端以外，这个情况非常容易处理；如果出发点在链上挂的子树内，显然和在链上的情况是等价的（因为必须得出去）；在链上，每次以二分之一的概率往左，二分之一的概率往右。

设 $f_{i,j}$ 为左边还有 i 个点、右边还有 j 个点，先到达左边的概率是多少。那么答案是 $\sum \frac{sz_i}{\sum sz} \cdot f_{i,len-i}$。

我们直接预处理出所有这些，然后三次方做即可通过。

直观上可能会认为 f 是一次函数，但是这是错的。它的差分类似于一个横过来的杨辉三角的形式。