IOI2020 中国国家集训队第一阶段作业 解题报告

西北工业大学附属中学 李佳衡

cf611G New Year and Cake

题目大意

给一个 n 个点的严格凸多边形，在 $\frac{n(n-3)}{2}$ 条对角线中任选一条切开，求所有方案中两部分面积差之和，答案乘 2 后对 10^9+7 取模。

数据范围

$4 \le n \le 5 \times 10^5$，顶点坐标为 $[-10^9, 10^9]$ 中的整数。

解题过程

设多边形的顶点逆时针依次为 $p_1, p_2, \ldots, p_n$，再设 $p_i = p_{i-n}\ (n < i \le 2n)$。

设 $A(l,r)$ 表示 $p_l, p_{l+1}, \ldots, p_r$ 组成的多边形的面积的二倍，特别地当 $r - l \le 1$ 时定义为 0，则有：

$$A(l,r) = \sum_{i=l}^{r-1} p_i \times p_{i+1} + p_r \times p_l$$

其中 $\times$ 表示向量叉积，即 $(x_1, y_1) \times (x_2, y_2) = x_1 y_2 - x_2 y_1$。

则答案即为

$$\begin{aligned}
&\sum_{l=1}^{n} \sum_{r=l+2}^{l+n-1} \left[A(l,r) \le \frac{A(1,n)}{2}\right] (2A(1,n) - A(l,r)) \\
=&2A(1,n) \sum_{l=1}^{n} \sum_{r=l+2}^{l+n-1} \left[A(l,r) \le \frac{A(1,n)}{2}\right] - \sum_{l=1}^{n} \sum_{r=l+2}^{l+n-1} \left[A(l,r) \le \frac{A(1,n)}{2}\right] A(l,r)
\end{aligned}$$

枚举对角线的一个端点 $p_l\ (1 \le l \le n)$，则一定存在一个 $r_l\ (l+1 \le r_l < l+n)$，使得对于任意的 $l+2 \le i < l+n$，$A(l,i) \le \frac{A(1,n)}{2}$ 当且仅当 $l+2 \le i \le r_l$。而由于 $A(l+1,r) \le A(l,r) \le A(l,r+1)$，有 $r_i \le r_{i+1}\ (1 \le i < n)$，可以双指针计算这个 r_l。

设

$$\begin{aligned}
S(l,r) &= \sum_{i=l}^{r} A(l,i) \\
&= \sum_{i=l}^{r} \left(\sum_{j=l}^{i-1} p_i \times p_{i+1} + p_i \times p_l \right) \\
&= \sum_{i=1}^{r-1} (r-i) p_i \times p_{i+1} + \left(\sum_{i=l}^{r} p_i \right) \times p_l
\end{aligned}$$

则有

$$S(l,r+1)-S(l,r)=\sum_{i=l}^{r}p_i\times p_{i+1}+p_{r+1}\times p_l$$
$$S(l+1,r)-S(l,r)=-(r-l)p_l\times p_{l+1}-\left(\sum_{i=l}^{r}p_i\right)\times p_l+\left(\sum_{i=l+1}^{r}p_i\right)\times p_{l+1}$$

我们记录

$$s_1(l,r)=\sum_{i=l}^{r}p_i$$
$$s_2(l,r)=\sum_{i=l}^{r-1}p_i\times p_{i+1}$$

则 $s_1(l,r)$ 和 $s_2(l,r)$ 都可以在 l,r 移动时 $O(1)$ 转移，因此 $S(l,r)$ 也可以 $O(1)$ 转移，答案即为：

$$2A(1,n)\sum_{l=1}^{n}(r_l-l-1)-\sum_{l=1}^{n}S(l,r_l)$$

总时间复杂度 $O(n)$。

agc034d Manhattan Max Matching

题目大意

对于 $1\le i\le N$，在 (RX_i,RY_i) 处放 RC_i 个红球，在 (BX_i,BY_i) 处放 BC_i 个蓝球，红球与蓝球总数相等，记作 S。定义一个红球与蓝球匹配的权值为其曼哈顿距离，求最大权满匹配。

数据范围

$1\le N\le 1000$，$0\le RX_i,RY_i,BX_i,BY_i\le 10^9$，$1\le RC_i,BC_i\le 10$，输入都为整数。

解题过程

对于两个球 (rx,ry) 与 (bx,by)，其权值一定一下四种之一：

1. $(rx-bx)+(ry-by)=(rx+ry)+(-bx-by)$
2. $(rx-bx)+(-ry+by)=(rx-ry)+(-bx+by)$
3. $(-rx+bx)+(ry-by)=(-rx+ry)+(bx-by)$
4. $(-rx+bx)+(-ry+by)=(-rx-ry)+(bx+by)$

实际权值即为以上四个的最大值，因此选择错误的式子一定不优，可以随便匹配，因此只需满足选择每种式子计算贡献的红、蓝球数相等。

可以使用费用流解决，即：

- 源点向每个红球代表的点连边，费用为 0，流量为 RC_i
- 每个红球代表的点分别向代表四种算式的点连边，费用为对应的贡献，流量为 $+\infty$
- 每种算式代表的点向每个蓝球代表的点连边，费用为对应的贡献，流量为 $+\infty$
- 每个蓝球对应的点向汇点连边，费用为 0，流量为 BC_i

答案即为最大费用最大流的费用，可以使用 Johnson 算法优化，时间复杂度 $O(N^2+SN\log N)$。

agc024f Simple Subsequence Problem

题目大意

给由长为不超过 N 的 01 串组成集合 S 和一个整数 K，求出最长的、是 S 中 K 个及以上字符串子序列的字符串，如果存在多个，输出字典序最小的。

数据范围

$0 \leq N \leq 20$，$1 \leq K \leq |S|$，所有长度不超过 N 的 01 串都可能同时出现在 S 中。

解题过程

对于两个字符串 s_1, s_2，如果要求 s_1 是 s_2 的子串，我们可以贪心地匹配，即对于 s_1 的每一位，向后找 s_2 中最靠前的与该位相同的位置匹配。

我们设 f_{s_1,s_2} 表示已经确定了 s_1，剩下的部分在 s_2 的子序列中确定的合法起点数（其中起点即为 S 中的字符串），初始时对于每个 $s \in S$，$f_{\emptyset,s} = 1$，其中 $\emptyset$ 表示空串。

转移时枚举 s_1 下一位选择 `0` 或 `1`，贪心地在 s_2 中删去该字符最早出现的位置及其之前的位置，可以向 $f_{s_1+c,s_2'}$ 转移，这样能转移到的所有状态都有 $|s_1| + |s_2| \leq N$，总状态数只有 $O(N2^N)$ 级别，转移可以做到 $O(1)$，总时间复杂度 $O(N2^N)$。