ftiasch's Contest 5

A. Even Three is Odd

设 $f(i,j,k)$ 表示已经确定了 $x_1, x_2, \ldots, x_{i+j}$，要求 $\max\{x_i, x_{i+1}, x_{i+2}\} = x_{i+j} = k$ 的方案数。其中，$0 \le j < 3$. 当 $f(i,j,k)$ 向 $f(i+1, j', k')$ 转移时，$x_{i+j \ldots i+1+j'}$ 的上界是 $\min\{k, k'\}$，拆开最小值部分和转移即可。复杂度是 $O(n^2)$.

B. Walk of Length 6

按照边集 $\{\{x_1, x_2\}, \{x_2, x_3\}, \ldots, \{x_6, x_1\}\}$ 分类，共有 9 种不同的图 $G_1, G_2, \ldots, G_9$. 分别计算 G 中同构于 G_i 的子图数量 $c_G(G_i)$，答案是 $\sum_{i=1}^{9} k(G_i) c_G(G_i)$。其中，$k(G_i)$ 是得到 G_i 的轨迹数量。

设 $\deg(u)$ 是点 u 的度，$\mathrm{common}(u,v)$ 是点 u, v 的公共邻居数量。后者可以通过 `std::bitset` 在 $O(\frac{n^3}{w})$ 得到。

1. $c_G(G_1) = \frac{\sum_{\{u,v\}\in E} \mathrm{common}(u,v)}{3}$;
2. $c_G(G_2) = \frac{\sum_{u<v} \binom{\mathrm{common}(u,v)}{2}}{2}$;
3. $c_G(G_3) = |E|$;
4. $c_G(G_4) = \sum_u \binom{\deg(u)}{2}$;
5. $c_G(G_5) = \left(\sum_{\{u,v\}\in E} (\deg(u)-1)(\deg(v)-1)\right) - 3c_G(G_1)$;
6. $c_G(G_6) = \sum_u \binom{\deg(u)}{3}$;
7. $c_G(G_7) = \sum_{\{u,v\}\in E} \binom{\mathrm{common}(u,v)}{2}$;
8. $c_G(G_8) = \sum_{u,v} \binom{\mathrm{common}(u,v)}{2} (\deg(u)-2)$;
9. $c_G(G_9) = \left(\sum_u \binom{\sum_v \mathrm{common}(u,v)[\{u,v\}\in E]/2}{2}\right) - 2c_G(G_7)$.

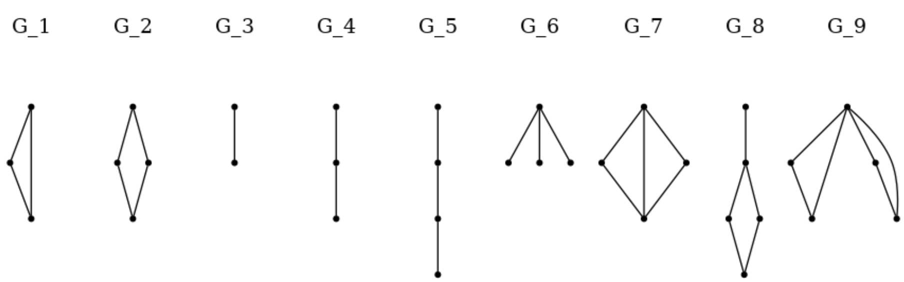

Figure 1:

C. City United

暴力状态压缩 DP 的复杂度是 $O(\mathrm{Bell}(\mathrm{pw}) \cdot m)$，无法通过。

考虑计算 $\sum_{S \subseteq V} 2^{c(S)} \bmod 4$，其中 $c(S)$ 是 S 的诱导子图的连通块数量。因为 $x > 1$ 时 $2^x \equiv 0 \pmod 4$，所以不连通的诱导子图贡献 0。连通的诱导子图贡献 2，所以答案是上式除以 2.

上式的直观解释是，选择顶点的子集 S，对 S 中的连通块黑白染色。交换求和号后等价于，把点染成黑、白、灰（代表不在 S 中的点）三种颜色，要求黑、白点不能相邻。

DP 的复杂度是 $O(3^{\mathrm{pw}} \cdot n)$.

D. Coins 2

令 $L = \mathrm{lcm}(1, 2, \ldots, n)$, $A = \sum_{i=1}^{n} a_i \cdot i$, 集合 $S = \{\sum_{i=1}^{n} x_i \cdot i : 0 \le x_i \le a_i\}$.

如果把 x_i 写成 $k_i \cdot \frac{L}{i} + r_i$ 的形式，那么 $S = \bigcup_{0 \le r_i < \frac{L}{i}} \{(\sum_{i=1}^{n} k_i) \cdot L + \sum_{i=1}^{n} r_i \cdot i : 0 \le k_i \le \lfloor \frac{a_i - r_i}{\frac{L}{i}} \rfloor\}$. 因为 $r_i \cdot i < \frac{L}{i} \cdot i = L$，所以 S 是若干公差为 L，首项不大于 $n \cdot L$ 的等差数列的并。又因为 $a_i - (\lfloor \frac{a_i - r_i}{\frac{L}{i}} \rfloor \frac{L}{i} + r_i) < \frac{L}{i}$，所以等差数列的末项不小于 $A - n \cdot L$.

总之，S 在 $[n \cdot L, A - n \cdot L]$ 中的数，以 L 为循环节。加上背包，总复杂度是 $O(n^2 L)$.

遗留问题：nL 是不是可以改进到 $2L$?

E. Lowest Common Ancestor

考虑轻重边剖分求 LCA 的过程。设 $\mathrm{head}(v)$ 表示点 v 所在重链的父亲，$\mathrm{head}^0(v) = v, \mathrm{head}^{k+1}(v) = \mathrm{head}(\mathrm{head}^k(v))$. 那么 $\mathrm{LCA}(u, v) \in \{\mathrm{head}^k(u) : k \in \mathbb{N}\} \cup \{\mathrm{head}^k(v) : k \in \mathbb{N}\}$.

计算 $f(i) = \sum_{1 \le j < i} w(\mathrm{LCA}(i, j))$ 时，按照 $\mathrm{LCA}(i, j)$ 分为两类，即

$$f(i) = \left(\sum_{1 \le j < i} w(\mathrm{LCA}(i,j))[\mathrm{LCA}(i,j) \in A] \right) + \left(\sum_{1 \le j < i} w(\mathrm{LCA}(i,j))[\mathrm{LCA}(i,j) \in B \setminus A] \right)$$

其中 $A = \{\mathrm{head}^k(i) : k \in \mathbb{N}\}$, $B = \{\mathrm{head}^k(j) : k \in \mathbb{N}\}$.

计算左式时枚举 $x = \mathrm{head}^k(i)$，统计 x 子树内的 j 的数量。计算右式时对每个 $j < i$，在 $y = \mathrm{head}^k(j)$ 处加上 $w(y)$，并查询 $(\mathrm{head}^k(i), \mathrm{head}^{k+1}(i))$ 链上的值。以上问题只需要 DFS 序上树状数组，复杂度 $O(n \log^2 n)$.

F. Multi-stage Marathon

设 $E(t, u)$ 是 t 时刻点 u 的期望人数，按照出发时间分段处理。对于每段，$E(t+1, \cdot) = E(t, \cdot) \times P$. 直接转移复杂度是 $O(n^3 T)$.

对所有 $0 \le C < n$，$O(n^4)$ 预处理 $E(t,u)$ 对 $E(t+C,v)$ 的贡献，即 $E(t+C,v) = \sum_{u=1}^{n} k_{C,u,v} E(t,u)$.

对于长度为 L 的段，先利用预处理的 k，$O(n^3)$ 得到 $E(t,n), E(t+1,n), \ldots, E(t+n-1,n)$ 的值，再 $O(n^3)$ 得到 $E(t+n,\cdot)$ 的值。一共进行 $\frac{L}{n}$ 轮，复杂度是 $O(n^3 \cdot \frac{L}{n}) = O(n^2 L)$.

整体复杂度是 $O(n^4 + n^2(T+m))$.

G. Matrix Recurrence

为了维护队列 Q，支持 `push_back` / `pop_front` 和询问所有元素的乘积，使用栈 F 和 B 来模拟 Q，满足 $\mathrm{rev}(F) + B = Q$.

其中 B 维护所有元素的乘积，F 维护所有到栈底的乘积。`push_back` 时往 F push，`pop_back` 时从 B pop。如果 B 为空，则把 F 翻转，重建得到 B.

因为任意元素只会进入 F, B 一次，所以复杂度是 $O(nm^3)$.

H. Permutation and noitatumreP

设 a_n 表示 n 的答案，则 $a_n = a_{n-1} + 2a_{n-3} + 2$.

详细证明 http://www.valpo.edu/mathematics-statistics/files/2014/09/Pudwell2015.pdf

I. Compressed LCS

设 $f(i,j)$ 表示 A 的前 i 段和 B 的前 j 段的 LCS，不妨假设 $c_i = d_j = \alpha$. 记 $p_{\alpha,i}$ 表示 A 的前 i 段中 α 的总长度，$q_{\alpha,j}$ 表示 B 的前 j 段中 α 的总长度。则

$$f(i,j) = \max_{0 \le i' < i, 0 \le j' < j} \{f(i',j') + \min\{p_{\alpha,i} - p_{\alpha,i'}, q_{\alpha,j} - q_{\alpha,j'}\}\}.$$

尝试去掉 $\min$，由对称性不妨假设 $p_{\alpha,i} - p_{\alpha,i'} \le q_{\alpha,j} - q_{\alpha,j'} \implies p_{\alpha,i} - q_{\alpha,j} \le p_{\alpha,i'} - q_{\alpha,j'}$. 然而以上限制不是十分有用，如果按照 j 递增枚举，则需要一个以 $(i, p-q)$ 为下标的数据结构，复杂度是 $O((n \log n)^2)$.

进一步观察，对于决策 (i',j')，只要满足 $p_{\alpha,i} - q_{\alpha,j} - b_{j+1} \le p_{\alpha,i} - q_{\alpha,j} \le p_{\alpha,i} - q_{\alpha,j}$，就可以取得 $p_{\alpha,i} - p_{\alpha,i'}$ 的贡献。所以只需要以 $(p-q)$ 为下标的线段树即可，复杂度是 $O(n^2 \log n)$.

J. Circle Sectors

根据格林公式，显然只需要知道最后图形的所有边界就好了。边界有两种：

1. 线段：不妨假设是 (x_1,y_1) 到 (x_2,y_2)，贡献是 $\frac{x_1 y_2 - x_2 y_1}{2}$；

2. 圆弧：不妨假设圆心在 (x_0, y_0)，半径是 r, 逆时针从 θ_1 绕到 θ_2，那么贡献是

$$\begin{aligned}
S &= \iint_D 1dxdy \\
&= \frac{1}{2}\oint_C xdy - ydx \\
&= \frac{1}{2}\int_{\theta_1}^{\theta_2} (x_0 + r\cos\theta)d(y_0 + r\sin\theta) - (y_0 + r\sin\theta)d(x_0 + r\cos\theta) \\
&= \frac{r}{2}\int_{\theta_1}^{\theta_2} [(x_0 + r\cos\theta)\cos\theta + (y_0 + r\sin\theta)\sin\theta]d\theta \\
&= \frac{r}{2}\int_{\theta_1}^{\theta_2} (x_0\cos\theta + y_0\sin\theta + r)d\theta \\
&= \frac{r^2(\theta_2 - \theta_1) + rx_0(\sin\theta_2 - \sin\theta_1) - ry_0(\cos\theta_2 - \cos\theta_1)}{2}
\end{aligned}$$

其实圆弧的贡献是：扇形面积 ＋ 两个有向三角形的面积，可以参考这个里面的图。

一种比较方便的求边界做法：枚举两个扇形 A 和 B，求出 A 和 B 之间的所有交点，然后交点把 A 的边界分成了若干段，判断每段的中点是不是在 B 内部就知道哪些段是有用的。然后对于每个 B 得出来的有用段，随便求个交集就好了。复杂度是 $O(n^2 \log n)$.

K. Welcome to ICPCCamp 2017

为了方便，假设 EC Final 的队伍排名是 $1, 2, \ldots, n$. 枚举最小的没有出线的队伍 i，那么 $Y = i - 1$，只需考虑 X 和 P 的影响。

设队伍 t 在区域赛中最高的排名是 $\mathrm{rank}(t)$，$\mathrm{count}_i(r)$ 满足 $t > i$ 且 $\mathrm{rank}(t) = r$ 的队伍 t 的数量。通过选择 X 和 P 能够得到的不同方案数是

$$1 + \sum_{r=1}^{\mathrm{rank}(i)} (2^{\mathrm{count}_i(r)} - 1).$$

只需从大到小枚举 i，通过树状数组维护询问即可。