[2021.10.10] 2021 CCPC Qualification Online

完成进度：A, B, C, J 还没看。

L. Contest Remake

> 给定 n 个**不能取**的正整数 $a_1, \cdots, a_n$ 和常数 C，求有多少个集合 s 满足：
>
> - 对于 $\forall i$，有 $s_i \in Z^+$；
> - 对于 $\forall 1 \le i < j \le |s|$，有 $s_i \ne s_j$；
> - $\prod\limits_{x \in s} x \le C$;
> - $s \cap \{a\} = \emptyset$。
>
> $1 \le T \le 20, 0 \le n \le 10^5, 1 \le C \le 10^9, 1 \le a_i \le C$，保证 a 互不相同。
>
> 保证 $\sum n \le 7 \times 10^5$，至多 10 组数据满足 $C > 10^6$。
>
> 时间限制 8000ms，空间限制 512MB。

Solution

我们设定一个阈值 B，若 $\max\limits_{x \in S} x < B$ 则直接爆搜，下面考虑 $\ge B$ 的情况。

令 $f[i][j]$ 表示考虑了 $1 \sim i$，它们乘积为 j 的方案数，这里 $i, j \in [1, \frac{C}{B}]$。 转移复杂度是个调和级数的过程。紧接着有如下常数优化：

- 我们预先已经计算了 $< B$ 的情形，所以 $f[B-1][*]$ 可以直接查表，我们要 DP 的第一维只需 $B \sim i$。
- 爆搜，我们采用 meet-in-middle，分割成 $[1, S]$ 和 $[S+1, B]$ 两部分，然后合并，这样 B 可以设的更大。

这题我写的真的快吐血了......

[2021.11.07] 2021 CCPC Guilin Onsite (GP of EDG)

A, B, D, E, F, G, I, K 是 naive 题。

完成进度：C 貌似是个奇怪的 DP 然后用分块优化？**我还不会。**

C. AC Automaton

> 给定一棵树，点 i 上标有字符 $s_i \in \{\text{'A'}, \text{'C'}, \text{'?'}\}$，父亲是 p_i $(1 \le p_i < i)$。
>
> 有 q 次修改，每次将 s_x 修改为 $y \in \{\text{'A'}, \text{'C'}, \text{'?'}\}$，求将每个 $s_i = \text{'?'}$ 的改成 $\text{'A'}/\text{'C'}$，最大化
>
> $$\sum_{x\text{是}y\text{的祖先}} [s_x = \text{'A'} \& s_y = \text{'C'}]$$
>
> $1 \le n, q \le 300000$。
>
> 时间限制 8000ms，空间限制 512MB。

H. Popcount Words

> 我们定义区间 $[l,r]$ 的 popcount word 为：$w(l,r)=\overline{s_l s_{l+1}\cdots s_r}$，其中 $s_i=\mathrm{popcount(i)} \bmod 2$。
>
> 给定 n 个区间 $[l_i,r_i]$，记 $S=w(l_1,r_1)+w(l_2,r_2)+\cdots+w(l_n,r_n)$，$+$ 表示字符串的拼接。
>
> 现在有 q 次询问，每次给定 01 串 p，问 p 在 S 中出现的次数。
>
> $1\le n,q\le 10^5, 1\le l_i\le r_i\le 10^9, \sum|p|\le 500000$。
>
> 时间限制 $1000\mathrm{ms}$，空间限制 $512\mathrm{MB}$。

Solution

定义 $W_{n,0}=w(0,2^n-1), W_{n,1}=w(2^n,2^{n+1}-1)$，容易发现：

- $W_{n,i}=W_{n-1,i}+W_{n-1,i\oplus 1}\ (i=0/1)$;
- $w(k\cdot 2^n,(k+1)\cdot 2^n-1)=W_{n,\mathrm{parity(k)}}$。

我们将 $w(l,r)$ 分解为 $O(\log r)$ 个 W 的形式，具体操作为：从小到大枚举 n，再从大到小枚举 n，找到 $l=k\cdot 2^n$ 且 $l+2^n-1\le r$，提取出 $W_{n,\mathrm{parity}(k)}$，变成 $w(l+2^n,r)$ 的子问题。

我们将所有 p 建出 AC 自动机，那么一个串的答案即为 fail 树中它 end 位置的子树和。问题转化为求出每个节点被遍历的次数。

因此，令 $go_{u,n,i}$ 表示从点 u 出发，跳了 $W_{n,i}$ 串到哪个点。显然 $go_{u,n,i}=go_{go_{u,n-1,i},n-1,i\oplus 1}$。

接着，我们考虑一种“从高位到低位，逐位下传贡献”的思想：

令 $f_{u,n,i}$ 表示从 u 开始走 $W_{n,i}$ 的次数，则：

$$f_{u,n,i}=f_{u,n+1,i}+\sum_{g_{v,n+1,i\oplus 1}=u} f_{v,n+1,i\oplus 1}$$

可以发现转移是从 $n+1$ 位降到 n 位，最后降到 0 位。

那么，节点 u 被遍历的次数为 $f_{fa_u,0,0}/f_{fa_u,0,1}$（取决于 u 是 fa_u 的哪个儿子）。最后在 fail 树上 dfs 计算出子树和即可。

时间复杂度 $O((n+|p|)\log r)$。实现的时候将倍增的量（n 在 u 前）提前，寻址连续，常数会小。

J. Suffix Automaton

> 给定一个串 S，我们将它的所有本质不同子串按照**先长度再字典序**的优先级排序。
>
> 有 q 次询问，每次输出第 K 小子串对应原串第一次出现的位置。
>
> $1\le |S|\le 10^6, 1\le q\le 10^6, 1\le K\le 10^{12}$。
>
> 时间限制 $6000\mathrm{ms}$，空间限制 $512\mathrm{MB}$。

Solution (后缀树)

首先建出后缀树，我们可以 $O(n)$ 计算出每种长度的本质不同子串个数，问题转化为求长为 i 的第 k 小本质不同子串。

后缀树的每个节点的父边代表一个长度的区间，因此我们跑扫描线来处理询问。将后缀树按照字典序排成 dfs 序，然后用线段树维护每个节点是否还存活（即当前枚举的 i 是否在该长度区间内），求第 k 小子串用线段树二分查询。

时间复杂度 $\mathcal{O}((n+q)\log n)$。

Solution (后缀数组)

首先通过 SA 求出 height 数组，那么长度为 l 的本质不同子串即为 height 数组里值 $< l$ 的个数，再剔除长度 $< l$ 的后缀。

同样考虑扫描线，我们时刻维护集合 S 表示所有 height $< l$ 且对应后缀长度 $\geq l$ 的下标，查询第 k 小即 S 中的第 k 个元素，可以用 `pbds` 中的 `set` 直接维护。由于要计算最小下标，因此我们还得用一颗线段树求区间 min（这里的区间即为 height $\geq l$ 的连续段，点权即为原串中的下标，如果后缀长度 $< l$，直接将对应点权赋 ∞ 就好）。

时间复杂度 $\mathcal{O}((n+q)\log n)$。

L. Wiring Engineering

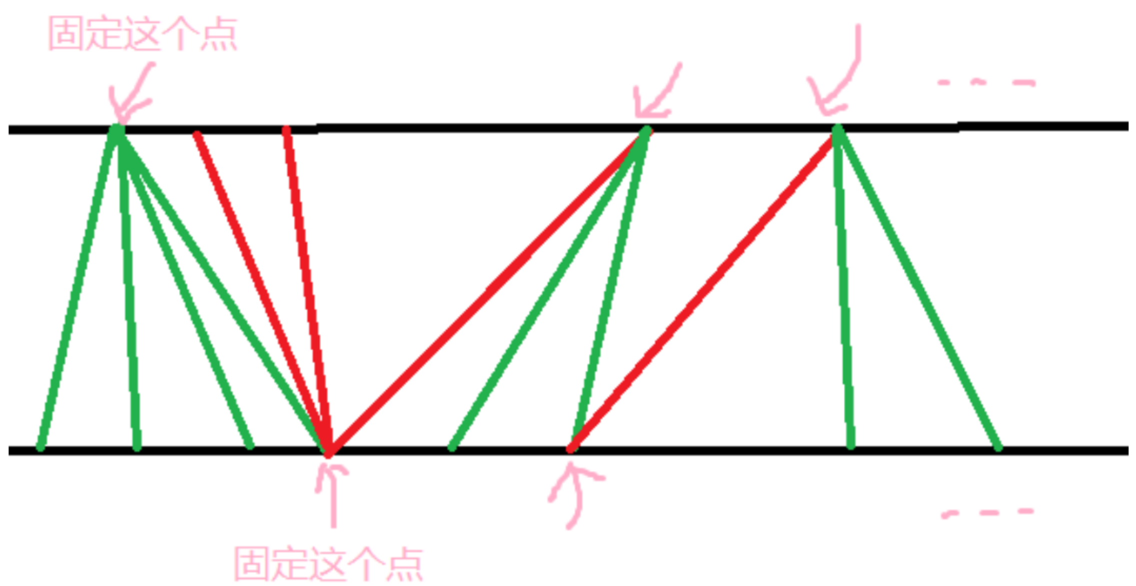
> 在字节街的北侧有 n 个建筑，第 i 个建筑的坐标为 $(i,1)$，南侧有 i 个塔，第 i 个塔的坐标为 $(i,-1)$。
>
> 你是一名电工，想要赚尽可能多的钱：
>
> - 如果连了一根从第 i 个建筑到第 j 个塔的电线，会赚 $w_{i,j}$ 美元。任意两根电线不能相交；
> - 如果你有至少一根电线连出第 i 个建筑，则要付 u_i 美元；
> - 如果你有至少一根电线连入第 j 个塔，则要付 v_j 美元。
>
> 现在有 q 次询问，每次给定 a,b,c,d，表示只有 $[a,b]$ 的建筑和 $[c,d]$ 的塔可用，求最大赚钱数。
>
> $1 \leq n \leq 500, 1 \leq q \leq 300000, 1 \leq u_i, v_i \leq 10000$。
>
> 时间限制 8000ms，空间限制 512MB。

Solution

先考虑朴素的 DP。定义：

- $f_{i,j}$ 表示只连接 $[a,i)$ 和 $[b,j)$ 的最优解；
- $g_{i,j}$ 表示只连接 $[a,i]$ 和 $[b,j)$，且用了 i 的最优解；
- $h_{i,j}$ 表示只连接 $[a,i)$ 和 $[b,j]$，且用了 j 的最优解。

我们分析图像的长相，一定是形如“八爪鱼”式：



容易得到三者的转移式：

$$
\begin{aligned}
f_{i,j} &= \max\{f_{i-1,j}, f_{i,j-1}, g_{i-1,j}, h_{i,j-1}\} \\
g_{i,j} &= \max\{f_{i,j-1} + w_{i,j-1} - u_i - v_{j-1}, g_{i,j-1}, g_{i,j-1} + w_{i,j-1} - v_{j-1}, h_{i,j-1} + w_{i,j-1} - u_i\} \\
h_{i,j} &= \max\{f_{i-1,j} + w_{i-1,j} - u_{i-1} - v_j, h_{i-1,j}, h_{i-1,j} + w_{i-1,j} - u_{i-1}, g_{i-1,j} + w_{i-1,j} - v_j\}
\end{aligned}
$$

初值 $f_{a,c} = 0$，答案即为 $\max\{f_{b+1,d+1}, 0\}$。时间复杂度 $O(qn^2)$。

我们观察到，DP 的转移形如在有向图上跑最长路（例如 $h_{i,j-1} \to g_{i,j}$，边权为 $w_{i,j-1} - u_i$），这启发我们从最长路的角度去分析它。记 $D_{i,j}$ 表示 i 到 j 的最长路。

考虑分治。令 k 为 $\frac{1+n}{2}$，我们此时处理满足 $a \le k, b \ge k$ 的询问。

在这里，因为 a, b 跨越 k，所以过程中必定经过 $h_{m,j}$（某个 j），因此：

$$D_{f_{a,c}, f_{b+1,d+1}} = \max_j \{D_{h_{k,j}, f_{a,c}} + D_{h_{k,j}, f_{b+1,d+1}}\}$$

于是对于 $j = 1 \sim n$，分别求出从 $h_{k,j}$ 出发到所有 f 的最长路。

算法的复杂度为 $T(n) = 4T(\frac{n}{2}) + O(n^3)$，即 $O(n^3)$。

实现的时候可以隐式建边，节省空间和常数因子。

[2021.11.14] 2021 CCPC Guangzhou Onsite

$A, C, F, G, H, I, J, K, L$ 是 naive 题。

完成进度：

D 大概会了，不过没细想过。
E 据说可以万欧？我**还不会**。
G, H, K, L 会了，不想写。

B 题有些麻烦，在笛卡尔树上用李超树维护线段。。

D. Unnamed Easy Problem

> idea $\in$ problem $\in$ problemset。
>
> 现在有 n 个 idea $0, 1 \cdots, n-1$。
>
> 一个 problem 是一些 idea 的集合，其至少包含一个 $k, k+1, \cdots, n-1$ 中的 idea。
>
> 现在无序地选择 m 个不同的 problem 组成 problemset，要求
> $0, 1 \cdots, o_{ee} - 1 \;\; k, k+1, \cdots, k + o_{ez} - 1$ 这些 idea 出现了奇数次，其余出现偶数次。
>
> 求合法的 problemset 个数，方案数对 19260817 取模。
>
> $1 \le k \le n \le 10^9, 1 \le m \le 10^6, 0 \le o_{ee} \le k, 0 \le o_{ez} \le n - k$。

Solution

考虑用生成函数去刻画它。

记奇数次 idea 集 $L = \{0, 1, \cdots, o_{ee} - 1, k, k+1, \cdots, k + o_{ez} - 1\}$。

记前半 idea 集 $B = \{0, 1, \cdots, k-1\}$。

用 x 表示集合幂级数（idea 集合），y 表示形式幂计数（个数）：

$$ans = [x^L y^m] \prod_{T \not\subseteq B} (1 + x^T y)$$

令 $F = \mathrm{FWT}(\prod_{T \not\subseteq B} (1 + x^T y))$，则 $F = \sum_S x^S \prod_{T \not\subseteq B} (1 + (-1)^{|S \cap T|} y)$。

再 IFWT 回去：

$$
\begin{aligned}
ans &= 2^{-n}\sum_T [x^T y^m](-1)^{|L\cap T|}F \\
&= 2^{-n}\sum_T (-1)^{|L\cap T|}[y^m]\prod_{S\not\subseteq B}(1+(-1)^{|S\cap T|}y)
\end{aligned}
$$

该式的前半部分：

$$
\sum_T (-1)^{|L\cap T|} = \begin{cases} \begin{cases} 2^n-2^{n-k} & L=\emptyset \\ 0 & L\not\subseteq B \\ -2^{n-k} & L\subseteq B \end{cases} & T\cap B\neq\emptyset \\ -1+[L\subseteq B]2^{n-k} & T\cap B=\emptyset \end{cases}
$$

该式的后半部分：

$$
\prod_{S\not\subseteq B}(1+(-1)^{|S\cap T|}y) = \begin{cases} (1+y)^{2^n-2^k} & T=\emptyset \\ (1-y)(1-y^2)^{2^{n-1}-1} & T\neq\emptyset \text{ and } B=\{0\} \\ (1-y^2)^{2^{n-1}-2^{k-1}} & T\neq\emptyset \text{ and } B\neq\{0\} \text{ and } T\cap B\neq\emptyset \\ (1-y)^{2^k}(1-y^2)^{2^{n-1}-2^k} & T\neq\emptyset \text{ and } B\neq\{0\} \text{ and } T\cap B=\emptyset \end{cases}
$$

前三项都可以组合数 $O(1)$ 计算，第四项直接进行 $O(mod)$ 的枚举前面的指数，推出后面的系数。

E. Mathlab

不会做。

[2021.11.21] 2021 CCPC Weihai Onsite

A,C,D,E,F,G,H,I,J,K,M 都是 naive 题。

B. Subset

> 求从 $[0,n]$ 中选 K 个不同的数，使得异或和的 popcount 为 B 的方案数。
>
> $1\le n\le 10^9, 1\le K\le 5000, 0\le B\le 30$。

Solution

先考虑 n 为 2 的整次幂情形，不妨假设数之间有序。

令 $dp[i][x]$ 表示考虑了前 i 个数，异或和为 x 的方案数，考虑容斥：

$$
dp[i][x] = n^{\underline{i-1}} - dp[i-2][x](n-i+2)(i-1)
$$

初值 $dp[0][0]=1, dp[1][*]=1$。我们发现 $x\neq 0$ 的 dp 值相同，因此改装成 $dp[i][0/1]$ 表示 x 是否为 0 即可。

现在考虑一般情形下的 n，我们将其二进制分解。以 $n=28$ 为例：

它 $(11100)_2$被划分成 $[0,16),[16,24),[24,28),[28,28]$，我们从低位往高位做，每次其实就是一个背包合并的过程。

假设当前已经处理到末 bit 位，设 $f[i][j]$ 表示选了 i 个数，它们的 popcount $=j$ （只考虑末 bit 位）的方案数。

注意到当前 bit 位意味着能把末 bit 位从 00...0 到 11...1 取遍，也就是能全覆盖之前处理过的位，所以转移是平凡的。

背包合并的过程可以用 NTT 优化，时间复杂度 $\mathcal{O}(K\log K\log^2 n)$。

L. Shake hands

> 有 n 个小朋友站成一排，初始第 i 个位置上占 i 号人。有 m 次操作，每次给定 x，位于位置 x 和 $x+1$ 的人握手，同时交换位置。求一个最大团，使得两两人均握过手。
>
> $2\le n\le 2\cdot 10^5, 1\le m\le 2\cdot 10^5$。

Solution

我们考虑 原图最大团 = 补图最大独立集，显然若 $x<y<z$ 满足 x,y 没握手，y,z 没握手，那么必然 x,z 没握过手（因为不可能跨越握手）。在补图中的表现就是传递闭包。

因此 补图最大独立集 = 补图最小链覆盖 = n - 二分图最大匹配，我们对于 $deg\ge\sqrt{m}$ 的点直接贪心取，因为它们始终保证 $|N(A)|\ge|A|$，由 Hall 定理知必定可行。对于 $deg\le\sqrt{m}$ 的暴力跑匈牙利，用并查集快速找到未被访问过的点，时间复杂度 $\mathcal{O}(n\sqrt{n})$。

[2021.11.21] 2021 CCPC Shenyang Onsite

题解仍在施工中，这场还没做完。

完成进度：

C, D 没看。
K 算是 rprmq2 了吧，矩形加矩形查，用查询分块 + kdt + 历史区间加线段树即可，但我不想写；
M 直接上 SAM 求就行，但是我不想写。

K. Matrix Operations

> 给定一个 $n\times n$ 的矩阵 a，执行 n 次以下操作：
>
> 给定 x,y,z_1,z_2,z_3,z_4，然后：
>
> - 令 $w_1=\max\limits_{i<x,j<y}(a_{i,j})$；
> - 令 $w_2=\max\limits_{i<x,y\le j}(a_{i,j})$；
> - 令 $w_3=\max\limits_{x\le i,j<y}(a_{i,j})$；
> - 令 $w_4=\max\limits_{x\le i,y\le j}(a_{i,j})$；
> - 给 $a_{i,j}(i<x,j<y)$ 加 z_1；
> - 给 $a_{i,j}(i<x,y\le j)$ 加 z_2；
> - 给 $a_{i,j}(x\le i,j<y)$ 加 z_3；
> - 给 $a_{i,j}(x\le i,y\le j)$ 加 z_4。
>
> 输出 w_1,w_2,w_3,w_4。
>
> $2\le n\le 10^5$。
>
> 时间限制 15000ms，空间限制 512MB。

Solution

考虑将询问分块，设块长为 B，那么整个大矩形被划分为 $B\times B$ 的矩形。

现在需要做两件事情：

- 计算块之前矩形每个位置的最大值

这相当于 $\mathcal{O}(m)$ 次矩形加，我们考虑一维跑扫描线，另一维放在线段上，那么相当于就是前缀历史最大值，以及在矩阵新的一行时候，打上清空标记。

这里时间复杂度 $\mathcal{O}(m+B^2)$。

- 维护块内时矩形每个位置的最大值

这个显然用 K-D Tree 维护即可。

这里时间复杂度 $\mathcal{O}(B^2)$。

因此，总时间复杂度 $\mathcal{O}(\frac{n}{B}(n\log n+B^2))$，取 $B=\sqrt{n\log n}$，最优复杂度 $\mathcal{O}(n\sqrt{n\log n})$。

[2021.11.28] 2021 CCPC Harbin Onsite

C. Colorful Tree

> 给定一棵以 1 为根的树，第 i 个点的父亲是 p_i $(1\le p_i<i)$，你需要将叶子 u 染成 c_u 颜色。一次操作你可以给 u 子树里的所有叶子染成同一个颜色，一个叶子若被多次染色，则取最后一次。问至少操作几次才能实现。
>
> $1\le n\le 10^5, 0\le c_i\le n$，若 i 不是叶子则 $c_i=0$，否则 c_i 表示染的颜色。

Solution

先考虑一个朴素的 DP：$dp[u][i]$ 表示 u 的祖先们（不含 u）给子树遗留了颜色 i，将子树内的叶子解决掉的最少次数。

转移先不考虑 u 点的染子树操作：$dp[u][i]=\sum\limits_v \min(dp[v][0],dp[v][i])$。

现在附加上 u 的操作：$dp[u][0]=\min\limits_{k=1}^{n}(dp[u][0],dp[u][k]+1)$。

这样复杂度是 $O(n^2)$ 的，考虑优化：

我们注意到，对于 u 而言，不在其子树内出现过的颜色没区别，因此有用的 i 只有 $O(\text{size})$ 个，因此启发式合并即可。

在最终实现的时候，我们需要额外存一个全局加的懒标记，细节较多。

时间复杂度 $\mathcal{O}(n\log^2 n)$。

K. Wonder Egg Priority

> 给定一个长为 n 的序列 a，有 q 次询问，模数为 M：
>
> - 给定 l,r,k，你需要对于 $i\in[l,r]$ 让 $a_i\leftarrow k^{a_i}\pmod M$；
> - 给定 l,r，计算 $\sum\limits_{i=l}^{r}a_i\pmod M$ 的值。
>
> $1\le n,q,M\le 10^5, 1\le p_i\le 10^5, 1\le k\le 10^5$。
>
> 时间限制 1000ms，空间限制 1024MB。

Solution

首先，每个位置上的数形如 $c_1^{c_2^{c_3^{\cdot^{\cdot^{\cdot}}}}}$ 的形式，容易发现 $M\to\varphi(M)$ 经过 $\mathcal{O}(\log M)$ 轮就变成 1，因此只有前 $\log$ 个 c_i 有用。

沿用这一个性质，我们考虑一种类似珂朵莉树的均摊思想：对于每个点 i 记一个势能函数 $\Phi(i)$，表示 i 和 $i+1$ 在一个连续段内的势能次数。初始它的值为 $\log M$。每次一个修改，会使得 $l-1$ 和 r 两个点的势能恢复至 $\log M$，并将区间内的所有点势能 -1，势能 ≤ 0 的点消失。该操作可以用一个 set 实现，然后在线段树上打对应的覆盖标记即可。

容易发现，该流程时间复杂度为 $\mathcal{O}(q\log M\log n)$，再加上我们预处理是采取类似 BSGS 的光速幂预处理，这里时间复杂度 $\mathcal{O}(\log M\sqrt{M}M)$，可以通过本题。