2024/9/26 CQYC NOIp 模拟赛 题解

题解：龙猫（totoro）

> P10580

首先 $x \nmid y$ 一定无解，考虑 $t = \frac{y}{x}$，将其分解质因数 $t = \prod_k p_k^{\beta_k}$。

每个质因子之间的贡献时独立的，对于每个质因子 p，那么问题变成了每个数为 $p^0, p^1, \cdots, p^\beta$ 中的一个，要求 n 个数中钦定选择 p^0, p^β 的方案数。

直接求不是很好求，考虑容斥，求出没有限制的方案数，减去钦定 p^0 或 p^β 没有在 n 个数中出现的方案数，加上 p^0, p^β 都没有在数列中的方案数。

那么答案就是 $\prod_k ((\beta+1)^n - 2\beta^n + (\beta-1)^n)$

时间复杂度 $\mathcal{O}(T\sqrt{n})$。

题解：悠然（youran）

> P10634

$s_i > s_j \land t_i > t_j$，$s_i = s_j \land t_i = t_j$，$s_i < s_j \land t_i < t_j$ 条件太麻烦了，不妨用广义 KMP 求出仅满足 $s_i = s_j \land t_i = t_j$ 的子串（其实就是记录了 S 中每个位置该字符的上一个出现位置，特判一下首次出现的情况，然后直接 KMP），接下来我们只需考虑字母在这个字串的排名，扫一遍，看是否与其在 T 中对应的字母所在的排名相等即可。

时间复杂度 $\mathcal{O}(n|\Sigma|)$。

题解：拆除炸弹（youyou）

> CF687D

本题为区间版本的 [NOIP2010 提高组] 关押罪犯。这个削弱版的做法，就是把所有边从大到小排序，依次遍历这些边。遍历到一条边时，如果可以把两个端点已经在同一集合，答案就是这条边的权值，否则将 $(u, v+n), (u+n, v)$ 合并。

直接将询问区间提出来按原题做可以获得 25 分，快一点的就是先排好序在去枚举判断，少个 $\log m$ 获得 35 分，时间复杂度 $O(mq\alpha(n))$。

然后是一个搬题人认为比较启示正解的一个 Subtask。
因为 $l_i = 1$，所以考虑从左到右扫描右端点去维护答案。我们发现真正有用的边最多是 $n-1$ 条合并成功的边，和一条合并失败（即答案）的边。直接维护这些边，扫描即可。时间复杂度 $O(nm\alpha(n))$，常数较小。

最后正解做法。我们可以建一棵线段树维护其区间内的边按权值排序后的状态，合并信息时使用归并排序，则建树是 $O(m \log m)$ 的。为了查询控制量级，只保留有用的边，虽多一个 $\alpha(n)$，但是常数小很多。查询时直接线段树里查询合并即可。

建树的复杂度为的 $O(m \log m\, \alpha(n))$，查询为的 $O(nq \log m\, \alpha(n))$，如果实现不烂已经可以通过了，即使常数写得大也给了 90 分。

最后一个小优化：因为查询只有 3000 个，有用的端点只有 6000 个，离散化后再建线段树，可以将 $\log m$ 优化成 $\log q$，优化后 std 在 0.7s 左右，可以通过。

（原题暴力就能过，搬题人使用了缩小时限的方法解决了这个问题）

题解：赖教（lai）

> AT_joisc2020_q

首先把每个计划看成一个在二维平面上左端点为 (t_i,l_i)，右端点为 (t_i,r_i) 的线段 P_i。

显然一定要选一个 $l_i=1$ 的线段和一个 $r_i=n$ 的线段。

子任务3（每个军队都满足 $l_i=1$ 或 $r_i=n$）：

此时最多只会选两条线段，假设有 (t_i,l_i,r_i) 和 (t_j,l_j,r_j) 两条线段，且 $l_i=1,r_j=n$。

- 如果 $t_i\le t_j$，那么要满足 $r_i-(t_j-t_i)\ge l_j-1$。
- 如果 $t_i>t_j$，那么要满足 $l_j+(t_i-t_j)\le r_i+1$。

做法就很明显了，将 t 排序，动态开点线段树维护。

正解：

~~结论题~~

考虑将子任务 3 的做法扩展一下，如下图所示，定义两条线段 i,j 存在单向边 $i\to j$ 的条件为（其实和子任务 3 一样）：

- 如果 $t_i\le t_j$，那么 $r_i-(t_j-t_i)\ge l_j-1$。
- 如果 $t_i>t_j$，那么 $l_j+(t_i-t_j)\le r_i+1$。

猜测一种方案是合法的当且仅当：**所选的方案中存在一个 $l_i=1$ 的点能到达所选的方案中任意一个 $r_i=n$ 的点。**

画下图就可以发现这个结论是正确的。

那么这就是个最短路模型了，归纳一下 i 和 j 有边当且仅当 $|t_i-t_j|\le r_i-l_j+1$。边权为 c_j。

起点为所有 $l_i=1$ 的点，终点为所有 $r_i=n$ 的点。但边数是 $O(m^2)$ 的。

注意到边权是在附在点上的，熟悉 Dijkstra 的同学应该知道这样的图在一个点**入堆后就不用再入堆了**，因为第一次入堆一定就是最小值。

因此我们只需要支持对于堆顶 x，快速得到它能扩展的没有入过堆的点 y 即可。明显可以用线段树维护。

具体的维护方法很简单，实在不会可以看代码。时间复杂度 $O(m\log m)$。

城市
时间