A. DNA 序列

一些稍微复杂且不稳的做法：

- 使用 `map` 维护（可能 TLE）。
- 使用字典树维护（可能 TLE、MLE）。
- 随便找一个字符串散列算法，使用哈希表维护。
- 使用后缀树、后缀数组、后缀自动机 ... 维护（😅）。

但是这是本场模拟赛的送分题，肯定是没有这么复杂的。

注意到字符集中总共只有 4 个字符 `A, T, G, C`，那么最多只会出现 4^k 种不同的长度为 k 的串，并且 k 很小。

如果我们将 `A` 换成 `00`；将 `T` 换成 `01`；将 `G` 换成 `10`；将 `C` 换成 `11`。于是就可以将一个字符串替换为一个每个字符占两位的二进制数。总的状态数为 $4^k \le 1048576$，是非常可以接受的。

至于如何计算每个长度为 k 的子串的映射值，可以从头扫到尾计算，利用位运算将无用状态取出，并加入新状态。

时间复杂度 $\mathcal{O}(n+4^k)$。
空间复杂度 $\mathcal{O}(4^k)$。

B. 奶茶计划

题目可以简化为：维护一个集合，支持插入，删除，按时间撤销删除，查询 mex。

算法一

先来考虑一下只有插入，查询 mex 的情况。

易发现答案是单调不降的，可以使用一个桶来维护每个编号的奶茶是否出现过，维护一个表示答案的指针，每次加入后尝试修改这个指针即可。

时间复杂度 $\mathcal{O}(m)$。
空间复杂度 $\mathcal{O}(m)$。

算法二

现在考虑一下有了删除，按时间撤销删除操作时要怎么做。

我们同样维护出只考虑插入操作时的答案 x，然后记录一下被删除的元素中的编号最小值 y。
显然答案为 $\min(x,y)$。这样就可以把插入、删除分成互不干扰的两部分。

现在的任务就是要维护删除元素集合的编号最小值。

使用线段树、平衡树等数据结构可以做到 $\mathcal{O}(m\log m)$，但并不优秀。注意到题目中，撤销删除操作是按照时间进行的。对于任意两个元素 x,y，若 x 被删除的时间比 y 早，且 x 的编号比 y 大，那么此时元素 y 一定是更优的。于是我们就可以维护一个删除时间递增，编号递增的单调队列，时间复杂度 $\mathcal{O}(m)$
。

C. 小小网格

直接上莫比乌斯反演：

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{m}\varphi(\gcd(i,j)) &= \sum_{d=1}\varphi(d)\cdot\sum_{i=1}^{n}\sum_{j=1}^{m}[\gcd(i,j)=d]\\ &= \sum_{d=1}\varphi(d)\cdot\sum_{i=1}^{\left\lfloor\frac{n}{d}\right\rfloor}\sum_{j=1}^{\left\lfloor\frac{m}{d}\right\rfloor}[\gcd(i,j)=1]\\ &= \sum_{d=1}\varphi(d)\cdot\sum_{i=1}^{\left\lfloor\frac{n}{d}\right\rfloor}\sum_{j=1}^{\left\lfloor\frac{m}{d}\right\rfloor}\sum_{p\mid\gcd(i,j)}\mu(p)\\ &= \sum_{d=1}\varphi(d)\cdot\sum_{p=1}\mu(p)\cdot\left\lfloor\frac{n}{pd}\right\rfloor\cdot\left\lfloor\frac{m}{pd}\right\rfloor\end{aligned}$$

记 $T=pd$，则式子化简为：

$$\sum_{T=1}^{\min(n,m)}\left\lfloor\frac{n}{T}\right\rfloor\cdot\left\lfloor\frac{m}{T}\right\rfloor\cdot\left(\sum_{d\mid T}\varphi(d)\cdot\mu\left(\frac{T}{d}\right)\right)$$

设 $g=\varphi*\mu$（这里的 $*$ 是狄利克雷卷积，下文同），则式子化简为：

$$\sum_{T=1}^{\min(n,m)}\left\lfloor\frac{n}{T}\right\rfloor\cdot\left\lfloor\frac{m}{T}\right\rfloor\cdot g(T)$$

注意到函数 g 显然为积性函数，可以使用线性筛将 g 的前缀和筛出。

配合数论分块即可做到 $\mathcal{O}(n)$ 预处理、$\mathcal{O}(\sqrt{n})$ 回答询问。但还不够优秀。

可以考虑杜教筛，设：

$$S(n)=\sum_{i=1}^{n}g(i)$$

记 $d=1*1$，其中函数 $1(n)=1$，显然函数 d 为约数个数函数。

那么可以得到 $S(n)$ 有关 $S\left(\left\lfloor\frac{n}{d}\right\rfloor\right)$ 的递推式：

$$d(1)S(n)=\sum_{i=1}^{n}(g*d)(i)-\sum_{i=2}^{n}d(i)\cdot S\left(\left\lfloor\frac{n}{i}\right\rfloor\right)$$

化简：

$$\begin{aligned}d(1)S(n)&=\sum_{i=1}^{n}(\varphi*1*\mu*1)(i)-\sum_{i=2}^{n}d(i)\cdot S\left(\left\lfloor\frac{n}{i}\right\rfloor\right)\\ d(1)S(n)&=\frac{n\cdot(n+1)}{2}-\sum_{i=2}^{n}d(i)\cdot S\left(\left\lfloor\frac{n}{i}\right\rfloor\right)\end{aligned}$$

该式显然需要数论分块来做。取一个阈值 K，分段处理：

- 对于 $n\le K$ 的部分，使用线性筛求出每个 $S(n),\sum_{i=1}^{n}d(i)$。
- 对于 $n>K$ 的部分，使用数论分块求解 $\sum_{i=1}^{n}d(i)$，使用上述递推式求解 $S(i)$。

可以使用**积分近似**证明当 $K=\mathcal{O}(n^{\frac{2}{3}})$ 时，取到杜教筛的理论最优复杂度 $\mathcal{O}(n^{\frac{2}{3}})$，证明留给读者。

D. 星际大战

注意到边双联通分量里的所有点肯定是在同一个联盟中的，此题的任务就是要维护图中的所有边双联通分量。

不妨先维护出原图中的边、新加入的边中的生成森林。

使用并查集维护边双（特别要维护一个集合大小），重新考虑一下动态加边的过程，不妨设当前加的边为 (x, y)，此时可能会遇到以下几种情况：

1. x, y 在图中本来就不连通，则直接连边即可。
2. x, y 在图中联通，且属于同一个并查集，则说明 x, y 本来就属于同一个联盟。
3. x, y 在图中联通，且不属于同一个并查集，则说明 x, y 在没加边之前不属于同一个联盟；而加边了之后，在生成森林中 $x \to y$ 的路径上的所有点都属于同一个联盟，全都并入 $\mathrm{lca}(x, y)$ 所代表的并查集即可。

时间复杂度 $\mathcal{O}(n + m + Q)$。