1001 Static Query on Tree

解法一：

树链剖分，对于集合 A, B，将根到该点打上 a 标记 / b 标记；对于集合 C，将该点的子树打上 c 标记；最后统计同时有 3 种标记的节点个数。用线段树可以维护。

（我在验题之前甚至不知道树链剖分可以这么快秒掉这题，所以说，树剖是真的强大。）

解法二（离大谱的正解）：

考虑一个简化的问题 1，如果只有 A 的限制，也就是求一个集合里所有节点可以走到的节点数。首先对 A 根据节点在 dfs 序中的位置排序，然后将深度加起来，减去相邻节点 lca 的深度之和。

在考虑一个稍复杂的问题 2，如果只有 A, B 的限制。很显然这个问题即两个问题 1 的交集，但是直接算交集比较困难。我们有公式，两个集合大小之和等于交集大小加并集大小，因此只要算出 A, B, A ∪ B 在问题 1 的解，就能推断出并集的大小了。

最后就是原问题，其实就是分成了多个子树，在这些子树上进行问题 2 的操作。

1002 C++ to Python

签到，只要无视字母、下划线、冒号后输出即可。

1003 Copy

一个修改操作对后续的查询操作的影响：如果 $x \le r$ 就没影响，如果 $x > r$，相当于查询 $x - (r - l + 1)$，然后去掉修改操作。

因此可以离线，倒着处理所有修改操作，每个修改操作都让它之后的所有 $x > r$ 的询问 `x -= r - l + 1`。

但是这么处理还是 $O(n^2)$ 的。考虑到我们只要答案的异或和，就有，两个相同的查询可以抵消，因此同一位置至多只会查询一次。这样每个位置用 1 bit 的信息表示即可，也就是 bitset。

我们令 dp 第 i 位为 1 表示答案需要对 `a[i]` 异或。倒着遍历所有操作，如果是查询操作，`dp[x] ^= 1`，如果是修改操作，那么就让 r+1..n 这些比特右移 r-l+1，代码如下：

```
low = dp & (~bitset<N>(0) >> (N - r - 1));
high = dp & (~bitset<N>(0) << (r + 1));
dp = low ^ (high >> (r + 1 - l));
```

关于锅：

这题因为某些原因，暴力和正解的差距并不是很大甚至暴力可以比正解还快。出题人并没有验全面导致该题难度骤降，非常抱歉。

1004 Keychains

显然关键问题是如何求圆 A 和 圆 B 所在平面 的两个交点。得到这两个交点后，只要判断是否一个点在圆 B 内部，一个点在圆 B 外部（到圆心的距离小于半径 or 大于半径），就是答案了。

直接算圆和平面交点似乎不太好做，考虑到这两个交点肯定在两个圆所在平面上，因此先求这两个平面的交线，然后求交线和球 A 的交点（假设球 A 的圆心和半径就是圆 A 的半径）。

直线和球的交点的求法，和平面几何的直线和圆交点的求法类似，不细讲了。剩下的都是板子的事情了。

1005 Slayers Come

显然，每个技能都可以击败一个区间的野怪，我们先处理出每个技能可以击败的区间。
先计算区间的右端点：将所有的 $a[i]-b[i+1]$ 从大到小排序，再将所有的技能按 $R[j]$ 从大到小排序，依次扫描每个技能（$R[j]$ 相同的一起处理）。对于 $a[i]-b[i+1]\geq R[j]$ 的所有 i，则用并查集将 i 和 $i+1$ 合并。第 j 个区间的右端点就是 $X[j]$ 所在连通块的最右值。
左端点同理。

接下来就是区间重复覆盖计数问题：n 个位置，m 个区间，求选出的区间能够重复覆盖 $[1,n]$ 的方案数。
设 $dp[i]$ 表示恰好覆盖了区间 $[1,i]$ 的方案数。
我们将所有区间按照右端点从小到大进行排序，依次扫描每个区间，考虑一个区间 $[l,r]$ 对 dp 数组的贡献。
$$dp[r]+=\sum_{j=l-1}^{r}dp[j]$$
$0\leq i\leq l-2, dp[i]*=2$
用单点加法，区间乘法，区间求和的线段树维护 dp 数组即可。

1006 Bowcraft

令 $dp[i]$ 表示从 0 级升级到 i 级期望的数量。

假设现在等级为 i，买了一本书 (a,b)，记 $\alpha=\frac{a}{A},\beta=\frac{b}{B}$ 。

若使用这本书，升到 $i+1$ 级的期望是 $dp[i]+1+(1-\alpha)(1-\beta)\cdot(dp[i+1]-dp[i])+(1-\alpha)\beta\cdot dp[i+1]$

若不使用这本书，升到 $i+1$ 级的期望是 $dp[i+1]+1$

得到 dp 方程：

$$dp[i+1] = \frac{1}{AB}\sum_{a,b}\min\{dp[i+1]+1, dp[i]+1+(1-\alpha)(1-\beta)\cdot(dp[i+1]-dp[i])+(1-\alpha)\beta\cdot dp[i+1]\}$$

对于当前的等级 i 和一本书 (a,b)，如果要使用这本书，那么升到 $i+1$ 级，不使用的期望 $\geq$ 使用的期望。

即 $dp[i+1]+1 \geq dp[i]+1+(1-\alpha)(1-\beta)\cdot(dp[i+1]-dp[i])+(1-\alpha)\beta\cdot dp[i+1]$

化简，得 $dp[i+1] \geq dp[i]\cdot\frac{\alpha+\beta-\alpha\beta}{\alpha}$

从这个式子可以看出，一本书的 $\frac{\alpha+\beta-\alpha\beta}{\alpha}$ 越小越容易被使用。即 $\frac{\beta(1-\alpha)}{\alpha}$ 越小越好。

将所有书按照 $\frac{\beta(1-\alpha)}{\alpha}$ 从小到大排序，枚举所有的 t，只使用 $\frac{\beta(1-\alpha)}{\alpha}$ 值前 t 小的书，其他书不使用。则

$$AB\cdot dp[i+1] = (AB-t)\cdot(dp[i+1]+1) + \sum_{a,b(\text{前}t\text{小})} dp[i]+1+(1-\alpha)(1-\beta)\cdot(dp[i+1]-dp[i])+(1-\alpha)\beta\cdot dp[i+1]$$

化简，得 $dp[i+1] = \dfrac{AB+dp[i]\cdot\sum\limits_{a,b(\text{前}t\text{小})}\alpha+\beta-\alpha\beta}{t-\sum\limits_{a,b(\text{前}t\text{小})}1-a}$

枚举所有的 t，得到 $dp[i+1]$ 的最小值，递推可得 $dp[K]$。复杂度 $O(KAB)$

1007 Snatch Groceries

伪阅读理解题

tips: TL;DR: 意为 Too Long; Didn't Read

本文大部分定义以及描述摘自《Designing Data-Intensive Applications》Martin Kleppmann

题目问置信区间重叠前有几个人成功抢到菜。

按关键词为 $earliest$ 升序排序后，比较相邻两个，如果 $latest_i \geq earliest_i$ 则有重叠

证明。假设前面都没有重叠，那么 i 之后最有可能和 i 重叠的一定是 $earliest$ 最小的，反之如果最小的都没有重叠，那么之后必然不可能有一个区间与 i 重叠。所以这样排序处理没有问题。

1008 Keyboard Warrior

使用哈希算法来比较两个字符串是否相同。

我们对字符串进行压缩，把连续相同的字符压缩为 $\{ch,k\}$ 表示 k 个字符 ch。

可以用栈来维护当前缓冲区字符。

如果哈希函数是类似 $hash\_val(str) = str[1] \times P^{n-1} + str[2] \times P^{n-2} + \cdots + str[n] \times P^0$

那么根据等比数列求和公式可得 $hash\_val(ch, k) = ch \cdot \frac{1-P^k}{1-P}$

比较时方便起见，可以将首位和末位独立考虑，如果字符相同数量大于等于目标文本即可。

比较中间部分时可以不考虑 k 造成的总长度，直接取压缩后相同长度比较哈希值。

Q: 可以用 kmp 之类算法吗?

A: 因为有删除操作，一般的操作会造成反复横跳捏

1009 ShuanQ

$P \times Q \equiv 1 \mod M$

$P \times Q - 1 = k \times M, k \geq 1$

M 是 kM 的一个比 P, Q 大的质因子

最多只有一个质因子满足要求，如果有多个满足要求质因子 M_1, M_2 那么 $kM = M_1 \times M2 > P * Q$ 矛盾

1010 Assassination

不走回路且元器件最多，说明是最大生成树。

我们引入关键边和伪关键边，如果从图中删去某条边，会导致最小生成树的权值和增加，那么我们就说它是一条关键边。伪关键边则是可能会出现在某些最小生成树中但不会出现在所有最小生成树中的边。

如果存在关键边，只要在任意一条关键边安排一个杀手即可。 至于伪关键边，需要安排杀手使得无论如何都会被选到，问题转化为无向图最小割。

Reference 求关键边和伪关键边，考虑 Kruskal 算法，我们将边按边权从大往小顺序执行，那么当我们按照顺序处理完所有权值大于 w 的边之后，对应的并查集的连通性是唯一确定的，无论我们在排序时如何规定权值相同的边的顺序。我们将已经连通的点进行缩点，可以看作是同一个点。然后，我们同时处理所有权值等于 w 的边，会有三种类型的边：

1. 自环边：根据 Kruskal 算法，这样的边不会被添加进最小生成树中。
2. 桥边：当这条边被删去时，图的连通性就会发生改变。也就是说，这条边对于最小生成树而言是必须的，那么它就是关键边；
3. 非桥边：那么当这条边被删去时，图的连通性不会发生改变。也就是说，这条边对于最小生成树而言不是必须的，那么它就是伪关键边。

可以使用 Tarjan 算法求出桥边。再考虑伪关键边安排杀手时要分组讨论，把 w 相同且处于同一连通图的伪关键边分为一组，因为只有这些边才具有相互竞争关系，对所有伪关键边组求最小割，最后取最小值。

时间复杂度，排序 $O(m \log m)$，tarjan 每次都分出子图做，总复杂度 $O(n+m)$

求最小割使用网络流或 Stoer-Wagner 算法，$O(|V||E|+|V|^2 \log |V|)$ 其中每组 $|E|$ 不超过 10^2，而 $\sum |E| = M \leq 2 \times 10^5$，上限在 10^7 数量级且很难达到。总结：能过！

1011 DOS Card

一对匹配的值 $=$ 左 $*$ 左 $-$ 右 $*$ 右
线段树上维护以下8个变量：
区间最大值
区间次大值
区间最小值
区间次小值
选了一对的最大值
选了两对的最大值
（一对的值 $+$ 剩下的最大值）的最大值
（一对的值 $-$ 剩下的最小值）的最大值

1012 Luxury cruise ship

对于 N 较小的询问，用背包预处理出答案。
如果 N 较大，先用 365 将 N 减小到背包预处理的范围内，再算出答案。