线段树（segt）

原题：没找到

> 给一个序列 $a_1 \sim a_n$，按顺序给出 m 个区间加或询问区间和的操作。
>
> q 次询问给定 $1 \le x \le y \le m$，回答：若在原序列的基础上按顺序进行第 $x \sim y$ 个操作，所有询问操作的答案的总和是多少？
>
> 数据范围：$n, m, q \le 10^5$。

考虑拆询问：把 $[x, y]$ 内修改对 $[x, y]$ 内询问的贡献拆成如下几个部分：

- $[1, x-1]$ 内修改对 $[1, x-1]$ 内询问的贡献 C_1。
- $[1, y]$ 内修改对 $[1, y]$ 内询问的贡献 C_2。
- $[1, x-1]$ 内修改对 $[x, m]$ 内询问的贡献 C_3。
- $[1, x-1]$ 内修改对 $[y+1, m]$ 内询问的贡献 C_4。

那么答案就是 $C_2 - C_1 + C_4 - C_3$，其中 $a_1 \sim a_n$ 原始的贡献在 C_1 和 C_2 里就能计算。

C_1/C_2 直接模拟全过程即可，C_3 维护每个点被修改和询问覆盖的次数，每次增加 x 的时候增 / 删一个操作即可。

剩下的只有形如："前缀 $[1, c]$ 对后缀 $[d, m]$ 的贡献"的形式，考虑对前缀分块，拆成 $[1, k\sqrt{m}], [k\sqrt{m}+1, c]$ 两个部分（整块和散块）。

其中整块对后缀的贡献可以每次加入 $\sqrt{m}$ 个修改操作，然后逆序扫 $d = m \to 1$，每次根据前缀和 $\mathcal{O}(1)$ 计算并加入一个询问操作的贡献即可。

其中加入修改操作的部分用分块维护，$\mathcal{O}(\sqrt{n})$ 修改 $\mathcal{O}(1)$ 查询，做法类似树状数组，维护差分数组对查询的贡献即可。

接下来考虑散块对后缀的贡献，注意到对于每个询问，散块里的操作只有 $\mathcal{O}(\sqrt{m})$ 个。

倒序扫 $i = m \to 1$，每次加入第 i 个操作，然后对于 $d = i$ 的所有询问，暴力枚举对应散块里每个修改操作，$\mathcal{O}(1)$ 计算贡献即可。

注意到这一部分修改操作只有 $\mathcal{O}(m)$ 次，因此用分块根号平衡成 $\mathcal{O}(\sqrt{n})$ 修改即可。

时间复杂度 $\mathcal{O}((m+q)\sqrt{n} + m\sqrt{m})$。

保卫王国（protect）

原题：ROI 2016 D2T3，loj 3066

> n 个点的树上有 m 条路径，选两条路径最大化他们交集的大小，给出方案。
>
> 数据范围 $n,m\le 2\times 10^5$。

分类讨论路径交的形状：

- 若是祖先 - 后代链：那么枚举两条链交的底 u，那么所有一端在 u 子树内的路径都可以被选，我们选 LCA 深度最浅的两条更新答案，启发式合并维护即可。
- 若不是祖先 - 后代链：显然两条路径有相同的 LCA，对于每个可能的 LCA 分别讨论，考虑枚举路径交的其中一个端点 u，那么所有一端在 u 子树内的路径都可以被选，我们要选两条路径使得他们另一端的 LCA 尽可能的深，显然 dfn 越靠近的点 LCA 深度越大，那么我们按 dfn 排序用线段树合并维护所有相邻点对的贡献即可。

但是对于每个 LCA，都进行线段树合并复杂度是错误的，观察到我们只要在所有路径端点构成的虚树上运行此过程即可，因此每次建虚树，总复杂度变成 $\mathcal{O}(k\log n)$

时间复杂度 $\mathcal{O}(m\log^2 n)$。

维护 $\mathcal{O}(n\log n)-\mathcal{O}(1)$ LCA 并把第一类也用线段树合并维护可以做到 $\mathcal{O}(m\log n)$。

爬山（climb）

原题：EGOI 2021 D1T4 Luogu P9312

> 平面直角坐标系上有 n 个点，坐标为 $(1,h_1)\sim(n,h_n)$，保证 h 是一个 $1\sim n$ 的排列，对于横坐标相邻的点有线段连接，你的目标是沿着连接相邻两个点的线段遍历这 n 个点。
>
> 有 m 个区间，第 i 个区间 $[l_i,r_i]$ 有价格 w_i，需要在 p_i 上购买，你能从 h_i 移动到 h_{i+1} 当且仅当 $[h_i,h_{i+1}]$ 中的每个 x 都至少被一个你购买的区间覆盖。
>
> 对于每个区间 $[l_i,r_i]$ 求出：你购买 $[l_i,r_i]$ 后从 p_i 开始遍历所有点的最小花费。
>
> 数据范围：$n,m\le 2000$。

考虑设计 dp 状态：$dp_{\mathbf{S},i}$ 表示购买的线段覆盖了 $\mathbf{S}$，且当前位置在 i 时继续遍历所有点的最小花费，容易发现知道 $\mathbf{S}$ 和 i 就可以知道 i 能到达的整个区间，枚举下一步购买的线段即可。

注意到一个观察：对于一个横坐标的区间 $[l,r]$，想要拓展到 $(l,h_l)\sim(r,h_r)$ 中的所有点需要的 $\mathbf{S}$ 一定是一个包含 $[\min h_{l\sim r},\max h_{l\sim r}]$ 的连续区间 $[L,R]$，而剩余的线段可以在以后需要拓展时再购买，因此可以用 $dp_{L,R,i}$ 表示状态。

考虑进一步简化状态，注意到当我们确定覆盖区间 $[L,R]$ 后，能够拓展的横坐标区间 $[l,r]$ 可能有很多个不相交的段，因此我们需要记录一维 i 表示我们实际所在的是哪个段。

而注意到 L 一定是某个最小的 l_u，R 一定是某个最大的 r_v，因此当我们记录 u,v 后，横坐标区间 $[l,r]$ 一定经过 p_u,p_v，那么这样的 $[l,r]$ 就是唯一的，此时只需要记录状态 $dp_{u,v}$ 即可。

观察转移的形式：

$$
\begin{aligned}
dp_{u,v} &\leftarrow \min_{r_v<r_{v'}}\{dp_{u,v'}+w_{v'}\}\\
dp_{u,v} &\leftarrow \min_{l_{u'}<l_u}\{dp_{u',v}+w_{u'}\}\\
dp_{u,v} &\leftarrow \min_{l_k<l_u,r_v<r_k}\{dp_{k,k}+w_k\}
\end{aligned}
$$

即新购买的线段分别更新了左端点 / 右端点 / 同时更新 $[l_u,r_v]$。

考虑优化第一个转移，注意到一个结论：若 $r_i<r_j<r_k$，且 k 不能被 $dp_{u,j}$ 购买，那么 k 一定不能被 $dp_{u,i}$ 购买，因为 $[l_u,r_i]\subseteq[l_u,r_j]$，且表示的区间都包含 p_u，那么 $dp_{u,i}$ 能到的山峰集合 $\mathbf{I}$ 一定包含 $dp_{u,j}$ 能到的山峰集合 $\mathbf{J}$，若 p_k 不属于 $\mathbf{I}$ 则 p_k 一定不属于 $\mathbf{J}$。

因此我们对于一个确定的 u，对 v 按 r_v 从大到小排序处理，用小根堆维护最小的 $dp_{u,v'}+w_{v'}$，若当前的 $p_{v'}$ 无法到达则直接弹出，根据我们刚才的结论，$dp_{u,v'}$ 一定不可能更新 r_v 更小的状态。

对于第二种转移同理，我们对每个确定的 v 按 l_u 从小到大转移，分别维护小根堆即可。

考虑第三种转移：$dp_{k,k}\to dp_{u,v}$ 的过程，考虑用前两种转移来描述：$dp_{k,k}\to dp_{k,v}\to dp_{u,v}$，但问题是状态 $dp_{k,v}$ 是不合法状态（$l_k<l_v$），为了让这种转移可以被前两种转移刻画，我们定义这样的 $dp_{k,v}$ 为合法状态且值恰好是 $dp_{k,k}$ 即可。

按 l_u 从小到大枚举 u，按 r_v 从大到小枚举 v，维护 $2m$ 个小根堆分别处理 u,v 一定时的两种转移即可。

时间复杂度：$\mathcal{O}(m^2\log m)$。

绘画（draw）

原题：杭电多校 2021 R3 T12 hdu 6984

> 给一个长度为 n 的序列，每个位置 i 可以选择染色或不染色，选择染色可以有 a_i 种不同的颜色选择。
>
> 要求任意 $i-1,i$ 不能被同时染色，任意 $i-k,i$ 不能被同时染色，求合法方案数。
>
> 数据范围 $n \le 300$。

考虑重排整个序列，k 个数一行进行排列，容易发现原问题的限制变成不能有染色格子四联通，并且第 j 行的开头与第 $j-1$ 行的结尾不同色。

当 k 很小时，可以考虑一个 $\mathcal{O}(n2^k)$ 的算法，$dp_{i,S}$ 表示 i 的前 k 个位置的染色情况为 S 时的方案数，可以理解为维护轮廓线转移。

当 k 很大时，类似考虑轮廓线 dp，交换行列然后状压行的轮廓线 $dp_{i,S}$，但这要求第一行和最后一行交错后没有列被多次染色，因此我们需要枚举第一行的染色情况，时间复杂度 $\mathcal{O}(n4^{n/k})$。

简单根号分治可以发现复杂度为 $\mathcal{O}(n2^{\sqrt{2n}})$，显然无法通过，考虑优化，注意到任何时候轮廓线 S 都不能有相邻的 1（可能有一两个位置可以有相邻的 1，但可以忽略不计），因此大小为 k 的合法轮廓线 S 数量大概是 Fib_k 的，近似可以估计为 $((\sqrt{5}+1)/2)^k$，预处理出合法轮廓线集合即可做到，用这个观察去优化刚刚根号分治的两个做法即可。

时间复杂度 $\mathcal{O}(n((\sqrt{5}+1)/2)^{\sqrt{2n}}) \approx \mathcal{O}(n \times 1.618^{\sqrt{2n}})$。

数字表格（table）

原题：Topcoder SRM 700 Div1T3 Topcoder 14625

> 给一个 m 行的表格，第 i 行有 a_i 个位置，现有 n 个数字串 $S_1, \sim S_n$，依次等概率地插到某个未被占用的位置上。
>
> 若某个行 i 被填满，则过程立刻结束，产生的权值为这 a_i 个格子上填的数字串从左往右顺次拼接形成的十进制数。
>
> 求最终权值的期望。
>
> 数据范围：$n, m, |S_i| \le 300, \sum a_i < n+m$。

枚举哪一行在哪个时刻被填满：设填入 S_j 到第 i 行后第 i 行被填满：

考虑贡献：转成求每种填法的权值和，显然分母是 $\frac{1}{(\sum_{k=1}^{m} a_k)^{\underline{j}}}$，分子可以被拆成内外两部分，即 $X \times Y$，其中 X 表示 $j - a_i$ 个数插入 $1 \sim i-1, i+1 \sim m$ 这些行，且每行都没被填满的方案数，Y 表示 $1 \sim j-1$ 中选 $a_i - 1$ 个数与 S_j 拼成数字串，每种方案的权值和。

先考虑怎么求 X：考虑一个简单 dp，设 $f_{i,j}$ 表 j 个数插进 $1 \sim i$ 行符合题意的方案数，$g_{i,j}$ 表示 j 个数插进 $i \sim m$ 行符合题意的方案数，转移是简单的：

$$f_{i,j} = \sum_{k=0}^{a_i - 1} f_{i-1,j-k} \times a_i^{\underline{k}} \times \binom{j}{k}$$
$$g_{i,j} = \sum_{k=0}^{a_i - 1} g_{i+1,j-k} \times a_i^{\underline{k}} \times \binom{j}{k}$$

暴力 dp，注意到总转移量为：$\sum_{i=1}^{m} a_i \times \sum_{j=1}^{i-1} a_j \le (\sum_{i=1}^{m} a_i)^2 < (n+m)^2$，因此时间复杂度 $\mathcal{O}((n+m)^2)$。

最终的 X 直接拼合前后缀得到：

$$X = \sum_{k=0}^{j-a_i} \binom{j-a_i}{k} \times f_{i-1,k} \times g_{i+1,j-a_i-k}$$

然后考虑怎么计算 Y：

先从 $m=1$ 的情况入手，显然要拆位，求每个 S_i 的期望位权，枚举在 S_i 后面的点的集合 P，那么这种方案的贡献就是：

$$\frac{|P|! \times (a_1 - |P| - 1)!}{n!} \prod_{j \in P} 10^{|S_j|}$$

注意到系数只与 $|P|$ 有关，很容易用生成函数写出期望位权：

$$\frac{1}{n!} \sum_{k=0}^{a_1 - 1} k! \times (a_1 - k - 1)! \times [z^k] \prod_{j \ne i} (1 + 10^{|S_j|} \times z)$$

显然后面的生成函数可以直接回撤背包维护，时间复杂度 $\mathcal{O}((n+m)^2)$。

然后考虑一般的情况，考虑如何计算 S_j 的期望位权，我们还是钦定 k 个数排在 S_j 的后面，那么排在 S_j 前面的数直接用组合数选都是等价的，设 $F_{j-1}(z)=\prod_{k=1}^{j-1}(1+10^{|S_k|}\times z)$，得到期望位权为：

$$\sum_{k=0}^{a_i-1}k!\times(a_i-k-1)!\times\binom{j-k-1}{a_i-k-1}\times[z^k]F_{j-1}(z)$$

$F(z)$ 同样背包维护即可，然后考虑某个 $1\sim j-1$ 之间的 p，如何计算 S_p 的贡献，注意我们要分类讨论有没有钦定 S_j 在 S_p 后面，如果没有，组合数计算的时候要减掉强制选择的 S_j。

若不钦定 S_j，S_p 对答案总的贡献就是：

$$S_p\times\sum_{k=0}^{a_i-2}k!\times(a_i-k-1)!\times\binom{j-k-2}{a_i-k-2}\times[z^k]\frac{F_{j-1}(z)}{(1+10^{|S_p|}\times z)}$$

如果钦定了 S_j，S_p 对答案的总贡献就是：

$$S_p\times\sum_{k=1}^{a_i-1}k!\times(a_i-k-1)!\times\binom{j-k-1}{a_i-k-1}\times 10^{|S_j|}\times[z^{k-1}]\frac{F_{j-1}(z)}{(1+10^{|S_p|}\times z)}$$

注意到我们可以提前维护 $sum_k=\sum_p S_p\times[z^k]\frac{F_{j-1}(z)}{(1+10^{|S_p|}\times z)}$，每次先枚举 j，更新 $F_{j-1}(z)$，然后枚举 p，线性回撤背包后统计 sum_k，最后枚举 i 再按公式计算。

时间复杂度 $\mathcal{O}(nm(n+m))$。

拍卖会（auction）

原题：人造情感 Luogu P5642 | loj 6733

> 给你一颗 n 个节点的树，以及 m 条路径 (u,v,w)。一个路径集合 S 的重量 $W(S)$ 记为：找出 S 的一个权值之和最大的子集，该子集满足任何两条路径没有公共点，这个子集的所有路径权值之和就是 $W(S)$。
>
> 记 $c(u,v)=w$ 为最小的非负整数 w，使得对于给定的 m 条边组成的路径集合 U，$W(U\cup\{(u,v,w)\})>W(U)$。
>
> 求 $\sum_{u=1}^{n}\sum_{v=1}^{n}c(u,v)$ 对 998244353 取模后的结果。
>
> 数据范围：$n,m\le 3\times 10^5$。

其实 $c(u,v)$ 就是求删掉 $u\to v$ 路径上的所有点后，剩余树的每一部分的答案之和 W'，最后 $c(u,v)$ 就是 $W(U)-W'+1$。

我们可以把这个值拆成 f_u 表示 u 子树的答案，以及 g_u 表示删除 u 子树后外部的答案。

先考虑如何求 f_u：

- 若 u 没被覆盖，则贡献为 $\sum_{v\in \text{son}(u)} f_v$。
- 否则枚举一条 $\text{LCA}(x_i,y_i)=u$ 的路径，贡献是 z_i 加上删掉 $x_i\to y_i$ 后每个子树的 f_p 之和。

  考虑容斥，记 fs_u 表示 $\sum_{v\in \text{son}(u)} f_v$，那么答案就是 z_i 加上路径上每个点 fs_p 之和减掉 f_p 之和（要算上 fs_u 不算 f_u）。

路径求和问题可以用差分转成 $1\to x$ 的路径和，子树加单点查，树状数组维护。

然后考虑如何求 g_v（$u=fa(v)$）：

- 若 u 不选，则贡献为 $g_u+fs_u-f_v$。
- 否则考虑加入一条 $\text{LCA}(x_i,y_i)=u$ 的路径，设 w_i 为我们在上一个部分计算的这条路径的 z_i 加上切出来的每个子树 f_p 之和，那么我们用 $w_i+g_u-f_v$ 更新每个 g_v。

  观察一下这条路径对每个 p 的所有儿子的 g 影响，发现对于每个 p 至多有两个儿子的 g 不能更新，其他都能更新。

  - 若 0 个儿子不能更新：一定是 x_i,y_i 两个端点的所有儿子都能被更新，维护一个 ed_u 标记更新即可。
  - 若 2 个儿子不能更新：一定是一条 $x_i\ne u,y_i\ne u$ 的路径，此时 u 的儿子中不是 x_i/y_i 祖先的儿子都能被更新，倍增维护 x_i,y_i 在哪个子树即可。
  - 若 1 个儿子不能更新：不妨对每个 v 维护一个标记 ne_v，表示 v 的所有兄弟都能被 ne_v 更新。
    - 若 $x_i=u,y_i\ne u$，那么 $y_i\to u$ 上所有不是 u 的点的 ne 标记都能更新。
    - 若 $x_i\ne u,y_i\ne u$，那么 $x_i\to y_i$ 上所有不是 u 且不是 u 儿子的点的 ne 标记都能更新。

    注意到 ne 的更新都是路径修改，可以直接树剖，但我们观察到求 ne_v 时，更新过的路径的 LCA 都在 v 的子树外，因此有一个端点在 v 子树里的点一定经过 v。

    因此我们可以直接把路径要更新的值存在 x_i 和 y_i 上，求 ne_v 就是子树最大值，zkw 线段树维护。

    但注意到只有第一类路径能更新 u 儿子的 ne_v，因此先处理第一类路径然后处理 ne_v，最后处理第二类路径即可。

    维护的过程中我们还注意到 ed_u 其实就是线段树上 u 节点的值，因此直接在线段树上查点值即可。

最后得到了所有的更新操作后，可以直接建线段树维护每个点最终的 g_v，也可以找到更新权值最大的的一次操作先执行，然后把没被更新的 $\mathcal{O}(1)$ 个点暴力更新。

最后我们就得到了所有 f_u 和 g_u，类似容斥可得 $c(u,v)$ 就是路径上不为 LCA 的每个点的 $fs_u - f_u$ 加上 $g_{\mathrm{LCA}} + fs_{\mathrm{LCA}}$，拆贡献分别在 u, v, LCA 处统计对应的贡献即可。

时间复杂度 $\mathcal{O}((n+m)\log n)$。

迷宫（maze）

原题：HackerRank week of code 25 dag-queries

> 给定一张 n 个点 m 条边的 DAG，维护如下操作共 q 次：
>
> - 将以 u 为根的闭合子图权值赋值为 x。
> - 将以 u 为根的闭合子图权值与 x 取 $\min$。
> - 求 v 的权值。
>
> 数据范围：$n, m, q \le 10^5$，存在一组拓扑序 $1, 2, \dots, n$。

注意到闭合子图刻画是很难的，因此不能用常规的做法去刻画，考虑分块。

先考虑只有第一种修改操作的情况：对询问分块，$\sqrt{q}$ 一轮重构，重构时顺拓扑序 pushdown 下去，块内的维护连通性暴力更新即可。

但是注意到维护 DAG 连通性这个问题是很强的，只能 bitset 在 $\mathcal{O}\left(\frac{n^2}{\omega}\right)$ 时空复杂度内维护，但空间复杂度太大了，考虑对点的标号再分块，每次考虑 v 在一个区间内的询问，维护的时候也只维护所有点到这些 v 的连通性。

然后考虑怎么做第二种修改操作，注意到对于第 i 个询问，假设这个询问最后一次被覆盖是操作 j，那么我们只要求 $[j+1, i]$ 这个区间里第二类修改操作对 v 的影响，这又是一个分块，对整块预处理出所有操作对每个 v 的影响，散块用刚才一样的技巧优化空间复杂度暴力处理即可。

时间复杂度 $\mathcal{O}\left(\frac{n^2}{\omega} + (n+m+q)\sqrt{q}\right)$。