HDU 6391 Lord Li's problem

> 给定 n,k,v。
>
> 定义集合 $S=\{x \mid 0\le x<2^n, \operatorname{popcount}(x)=3\}$。
> 求满足 $T\subseteq S, |T|=k, \bigoplus_{x\in T} x=v$ 的集合 T 的数量。
>
> 对一个模数取模。
>
> $1\le n\le 40, 1\le k\le \min\{20,\binom{n}{3}\}, 0\le v<2^n$，$5\times 10^4$ 组数据。

首先因为顺序是不紧要的，但是发现如果就对于一个集合考虑会有点繁杂。
结合最后选的值一定不重，于是钦定选出来的是有顺序的，最后除掉 $k!$ 即可。

首先发现对于这个 v 实际上并不关心这些 $1,0$ 的值在哪里而是只关心个数，因为可以对最后的 01 建立双射一一映射上。

于是考虑设计 dp：$f_{i,j}$ 表示选了 i 个不同值，异或出的 1 的个数为 j 的方案数。
那么对于转移，就是考虑第 $i+1$ 个选的值，但是同样的因为只关心 1 的个数，所以对于这个值也只关心其有多少个 1 在这个值的 1 之中。
所以有 $f_{i,j}\times\binom{j}{c}\times\binom{n-j}{3-c}\to f_{i+1,j+3-2c}(0\le c\le 3)$。

此时还有一个问题是并没有解决选的不重这个问题。
不过根据状态设计，考虑到如果重了重的也只会是第 $i+1$ 个数和某个数。
而且这两个数在异或后就会抵消掉，于是可以当作是从 $i-1$ 个数的时候选了两个相同的且与原有的数不同的数出来。
还有一点是因为 dp 是考虑了顺序的，那么还要考虑把这个值插入进去，于是还有 $-f_{i-1,j}\times i\times(\binom{n}{3}-(i-1))\to f_{i+1,j}$。

时间复杂度 $\mathcal{O}(Tnk)$。

QOJ5357 芒果冰加了空气

> 给定一颗大小为 n 的树，求其点分树个数（每次选根节点不必为重心）。
>
> 对一个模数取模。
>
> $n \le 5000$。

考虑树上 DP。考虑已知两个子树的点分树方案数，求合并成一个子树的答案。但只知道 u, v 的编号信息显然是是不够的。

考虑对于任意一组方案，把两边的点分树给画出来。此时多加了 (u, v) 这条边，那么就在点分树上把 u, v 给找出来。在合并两边选择点分树的过程时，如果此时选了 u, v 的子树内的点，显然不影响树的形态。进一步分析，能够发现选的如果不是 u, v 到根路径上的点，那么顺序都是不影响的，因为其对应链上的点选了之后这些点就与 (u, v) 这条边没有关系了。

现在只需要考虑 $u \to rt_u, v \to rt_v$ 这两条路径，记 u, v 在点分树上的深度 d_u, d_v。那么 DP 就很好设计了，设 $f_{u,d}$ 表示 u 的子树内构成的点分树且 u 的深度为 d_u 的方案数。转移是简单的，枚举合并之后的 u 的深度就是一个组合数。

时空复杂度 $\mathcal{O}(n^2)$。

Luogu P11456 [USACO24DEC] Interstellar Intervals G

> 有一个长度为 n 的数轴，初始 n 个点均为白色。
>
> 定义一次操作为选择 $(i,x)(1\le i\le i+2x-1\le n)$，把 $[i,i+x-1]$ 涂成红色，把 $[i+x,i+2x-1]$ 涂成蓝色。
>
> 可以进行任意多次操作，但要满足任意两个操作选择的区间（$[i,i+2x-1]$）是不交的。
>
> 接下来给出了一个长度为 n 的由 `R`, `B`, `X` 组成的字符串。
>
> 问有多少种把 `X` 替换成 `R` 或 `B` 或 `W` 的方式使得该字符串可以从全白的初始状态操作得到。
> （`R` 代表红色，`B` 代表蓝色，`W` 代表白色）。
>
> 需要注意，数轴上的一些位置是可以不进行操作保持白色的。
>
> 对一个模数取模。
>
> $1\le n\le 5\times 10^5$。

对于这种问题，一个想法就是直接考虑前缀方案数，然后不断的拼一个区间上去。

所以考虑设 f_i 表示考虑了 $[1,i]$ 并且 i 被选中的方案数。
不过因为这题还有可能不选，于是再设一个 g_i 表示让 g_{i+1} 开头的方案数。

对于 g 的转移，就考虑这一位到底有没有被选：$g_i=f_i+[s_i=\texttt{X}]g_{i-1}$。
对于 f 的转移，就考虑前面开头在哪：$f_i=\sum\limits_{j=1}^{i}\operatorname{check}(j,i)g_{j-1}$。

那么瓶颈在于这个 f 的转移，继续分析发现主要是这个 $\operatorname{check}(l,r)$ 的问题。
于是考虑分析一下这个 $\operatorname{check}(l,r)$ 在什么时候为 1：

- $(r-l+1)\bmod 2=0$。
- 令 $mid=\lfloor\frac{l+r}{2}\rfloor$，则 $[l,mid]$ 能被染为 `R`,$(mid,r]$ 能被染为 `B`。
  但是这样不是很好判断，考虑反面：$[l,mid]$ 不存在 `B`，$(mid,r]$ 不存在 `R`。

于是发现对于一个 l，l 后面的第一个 `B` 就限制了其对应的 mid，就限制了对应的 r 的区间。
且对于一个 r，r 前面的第一个 `R` 就限制了其对应的 mid，就限制了对应的 l 的区间。
于是可以类似扫描线的维护这个贡献。

对于第一个条件，只需要分奇偶维护就行了。

时间复杂度 $\mathcal{O}(n\log n)$。

值得一提的是这题这个区间的限制很好看，所以是可以做到线性的。

GYM 102056E Immortal ... Universe

> 给定两个长度为 n, m 的由 +, -, ? 组成的字符串 $s_{1\sim n}, t_{1\sim m}$。
>
> 对于两个长度为 n, m 的由 +, - 组成的字符串 a, b，有如下定义：
>
> - 定义 $i=0, j=0, s=0$，“移动 i”操作需要满足 $i<n$，进行该操作后 $i \leftarrow i+1$，若 $a_i=+$，$s \leftarrow s+1$，否则 $s \leftarrow s-1$，“移动 j”操作则是在 b 上做相同的处理。
> - 若 $i=n, j=m$，该过程停止，否则有：
>   - 若 $s>0$，则会在满足条件的“移动 i”和“移动 j”的操作中随机一个进行。
>   - 若 $s=0$，则会在满足条件的“移动 i”和“移动 j”且满足进行该操作后 $s=1$ 的操作中随机一个进行，否则定义 (i,j) 是不通的。
>
> 若对于 (a,b)，不管怎样随机选择操作进行，都不会到达不通的状态，则称 (a,b) 是合法的。
>
> 现在询问由多少种把 s,t 中的 ? 替换成 + 或 - 后得到的新串 s',t' 满足 (s',t') 是合法的。
>
> 对一个模数取模。
>
> $1 \le n,m \le 5000$。

考虑刻画一下这个移动的过程。
观察一下，发现不能走当且仅当 $s=0$ 且走的是 -1。
于是会发现如果走过去 s 仍然能 ≥ 0 那么就是能走的。

同时会发现走到 (i,j) 时的 s 是确定的，即为 S_i+T_j（前缀和）。
于是考虑一个二维平面，(i,j) 能走当且仅当 $S_i+T_j \ge 0$。
那么条件就是不存在一个能从 $(0,0)$ 走到且不为 (n,m) 且两个方向都不能走的点。

经过一顿调整证明，这个就等价于：
给 s 在后面加一个哨兵 -1，t 也一样。不存在 $0 \le i \le n, 0 \le j \le m, (i,j) \ne (n,m)$ 使得 $S_i+T_j<0$ 且 $s_{i+1}=t_{j+1}=-1$。

于是考虑容斥不合法的情况。
那么为了不合法，一定会贪心的选择 $\min S_i, \min T_j (s_{i+1}=-1, t_{j+1}=-1)$。

于是考虑 dp，就需要在 dp 过程中维护这个最小值了。
于是一个 dp 想法就是 $f_{i,sum,min}$ 表示考虑到 $[1,i]$，当前前缀和为 sum，最小值为 min 的方案数。
不过这样子是 $\mathcal{O}(n^3)$ 的。

考虑优化一下。
考虑转移的时候钦定的 $+1,-1$ 影响的区间是一个后缀，这也是要同时维护 sum, min 的原因。
但是考虑倒着 dp，直接维护 $f_{i,min}$ 表示考虑了 $[i,n]$，最小值为 min 的方案数。
此时考虑转移钦定的 $+1,-1$ 影响的就是这一个后缀，就可以直接平移维护了。

同时需要注意的是还需要容斥一种情况：
$\min S_i$ 只在 $i=n$ 时取到，$\min T_j$ 只在 $j=m$ 时取到，且 $S_n+T_j=0$。
这部分跟上面的 dp 是类似的，只需要考虑转移时的一些条件就行了。

时间复杂度 $\mathcal{O}(n^2)$。

Luogu P11520 [THUPC 2025 初赛] 骑行计划

> 小 X 共有 n 天的假期，在第 i 天将会骑行 s_i 分钟，每分钟的骑行费用为 c 元。
>
> 为了节约开支，小 X 打算购买一些骑行卡。
> 现在有 m 种骑行卡可以购买，其中第 i 种骑行卡的具体信息如下：
>
> - 售价 w_i：每张卡的价格为 w_i 元；
> - 有效期 d_i：从购买当天算起，连续 d_i 天内有效；
> - 免费时间 t_i：在有效期内，每天的前 t_i 分钟骑行是免费的。
>
> 小 X 可以多次购买任意一种骑行卡，并且可以在同一时间持有多张有效的骑行卡。如果某天有多张骑行卡同时有效，那么当天可以享受的免费骑行时间为这些卡中 t_i 的最大值。对于超出免费时间的部分，仍然按照每分钟 c 元计算。
>
> 小 X 想知道在假期中骑行的最小总支出是多少。
>
> $1 \le n \le 150, 1 \le m, c \le 10^4, 1 \le s_i \le 150$。

注意这个取 t_i 的最大值，于是这启发取根据这个 t_i 的值域考虑。

一个想法可能是记 $f_{i,l,r}$ 表示只使用 $t_j \le i$ 的卡片，并且钦定已经取到了 $[l, r]$ 的最小花费的代价。
但是做一下会发现假了，因为可能存在一个 t_j 大一点 d_j 小一点的区间被一个 t_j 小一点 d_j 大一点的区间包围，此时这个贡献就计算不了了。

于是会发现只能从高到低的来 dp，也类似于是一个笛卡尔树逐层处理。
因为 t_i 大一点的一定能替换 t_i 小一点的，于是只保留 t_i 大一点的能产生正确的贡献的一定是一个区间的形式。
于是可以设计 $f_{i,l,r}$ 表示只使用 $t_j \ge i$ 的卡片，并且钦定已经取到了 $[l, r]$ 的最小花费的代价。

接下来就考虑扩展到 $i-1$，即去选取 $t_j = i-1$ 的卡片。
一个错误的转移方式就是直接拆分成许多个由 $t_j \ge i$ 和 $t_j = i-1$ 的卡片所覆盖的区间独立计算贡献。
问题还是和一开始一样，有可能这个 d_j 比较大，覆盖的比较长，单独计算贡献就算多了。

于是考虑在此时引入辅助 dp：$g_{l,r}$ 表示强制钦定 $[l, r]$ 一定被 $t_j = i-1$ 的卡片填满了的最小代价。
那么转移就可以考虑分讨：

- r 或许没有被覆盖过，那确实就是覆盖的最大值为 $i-1$。
- 存在 k 使得 $[k, r]$ 其实也被 $t_j \ge i$ 覆盖了，那么可以枚举这个 k，根据 $g_{l,k}$ 和 $f_{i,k+1,r}$ 来转移。

最后还需要让 $g_{l,r}$ 加上填上 $r-l+1$ 这个长度所需要的代价，可以背包。

那么此时就可以考虑把区间拆为被 $t = i-1$ 填了的区间和没有被 $t = i-1$ 填了的区间，即 g 和 f_i，对这个再做合并得到的 f_{i-1} 就对了。

时间复杂度 $\mathcal{O}(nm + n^3 \max t)$ 。

QOJ 6366 Message

> 给定两个字符串 S, T 以及长度为 $|S|$ 的数组 c_i，表示删去位置 i 的代价。
>
> 对于每个字符，你可以删去这个字符在 S 开头的若干个位置和结尾的若干个位置。
>
> 求最小花费使得 $S = T$ 或者报告无解。
>
> $|s| \le 2 \times 10^5, |\Sigma| = 26$。

对于每种字符最终只会保留一段区间。考虑把 T 中每个字符对应的区间拿出来，这对应到 S 上区间的相对顺序是不变的。

例如 $T = bbabca$，用 $L_b, L_a, R_b, L_c, R_c, R_a$ 来描述对应区间的关系。把 T 按照区间端点划分成若干部分：$|bb|ab|c|a|$。这个划分对于 S 是等价的，相当于要把 S 划分成若干部分，每个部分只保留字符集内部的所有字符，而且满足和 T 对应部分相同。

设 $dp_{i,j}$ 表示前 i 个数分成 j 段的最小代价。k 能够转移，当且仅当 $s[k+1, i]$ 在只保留若干字符之后恰好等于 $T[l_j, r_j]$。

考虑优化，首先我们对于 S 只保留当前需要的字符集合 C 得到 S'，然后用 S' 和 $T[l, r]$ 做一个匹配，对于每个匹配的位置，对应到 DP 上的转移形如一段区间的状态到另一段区间，而且花费为 $s_r - s_{l-1} + c$，其中 c 是一个常数，贡献独立，随便预处理一下就能做到单次 $\mathcal{O}(n)$。

最多只会有 $\mathcal{O}(|\Sigma|)$ 个段，故总的复杂度为 $\mathcal{O}(n|\Sigma|)$。

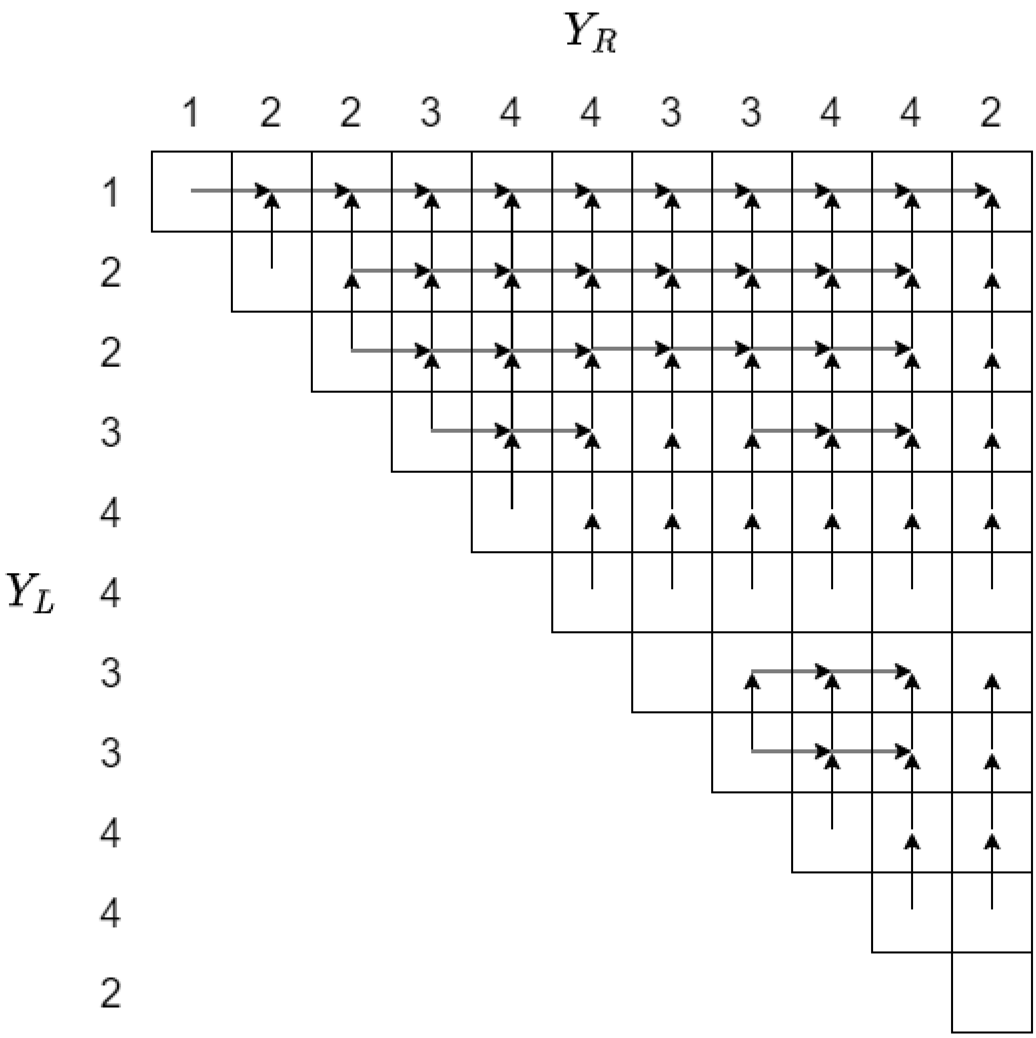
AT ARC153E Deque Minimization

> 对于一个十进制 X，对它进行如下操作：
> 有另一个十进制 Y，初始为空，从高到低对于每个 X_i，把它放到 Y 开头或结尾。
>
> 记 $f(X)$ 为 X 能操作出的最小的 Y。
>
> 现给定 Y，求有多少个 X 满足 $f(X)=Y$。
>
> 对一个模数取模。
>
> $1 \le Y < 10^{200000}$。

考虑对于一个 X ，$f(X)$ 的性质。容易发现，对于 X 中的前缀最小值，我们把它放到开头，否则放到结尾。设 $dp_{l,r}$ 表示有多少个 X 的前缀可以 $Y_{l,r}$，那么可以得到如下转移：

- 若 $Y_{l-1} \le Y_l$，则 $dp_{l,r} \to dp_{l-1,r}$。
- 若 $Y_{r+1} > Y_l$，则 $dp_{l,r} \to dp_{l,r+1}$。

画出转移图：



可以发现存在很多状态是没有意义的。

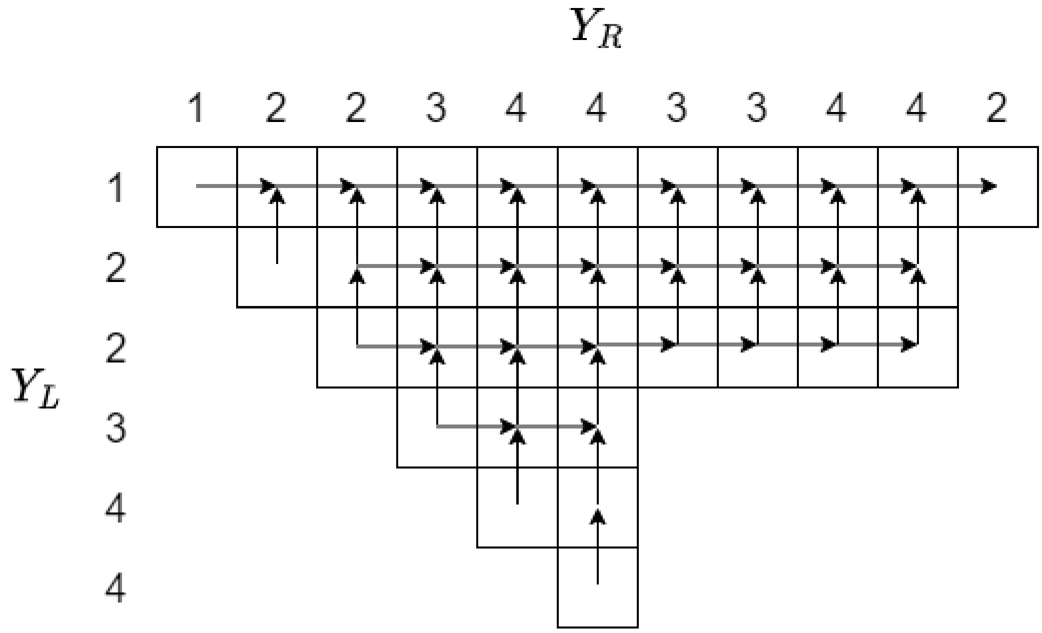
记 p 表示 Y 的最大不降前缀。

对于 l，那些 $l > p$ 的 $dp_{l,r}$ 不可能转移到 $dp_{1,n}$ ，这些状态可以去掉。

那么剩下的对答案有贡献的 l 满足 Y_l 单调不降。

对于 r，若 $r > p$，且存在一个 $k \in (p, r]$ 满足 $Y_k \le Y_l$，那么 $dp_{l,r}$ 不可能会被访问到，也可以去掉。

删去无用点后得到缩减后的图：



考虑从大到小枚举 l 转移。注意到 $1 \le Y_l \le 9$，考虑把 Y_l 相等的连续的 l 一起做，不难发现相等的部分建出来的图左边是若干个 1，右边就是一个矩形，每个点都可以往上或者往右走。

对于矩形部分，从 (l, r) 走到 $(l - x, r + y)$ 的方案就是组合数，即 $dp[l][r] = \sum_{r'} dp[l-x][r'] \times \binom{x+r'-r}{x}$，这是一个卷积形式，套用 NTT 优化至 $\mathcal{O}(9n \log n)$。

Luogu P10181 龙逐千灯幻

> 给定长为 n 的数组 a。
>
> m 次询问将前缀 $[1, x_i]$ 划分为 k_i 个区间，每个区间颜色数之和的最大值。
>
> $1 \le n \le 10^5, 1 \le m \le 10^6$。

设$dp_{i,k}$ 表示 $[1,i]$ 中划分了 k 个区间的最大美观度。

首先这个美观度 f 满足四边形不等式，因此 $dp_{i,k}$ 关于 k 是凸的。
由于 $dp_{i,k} \le i$，这个凸函数实际上还比较特殊：
设 $d_k = dp_{i,k} - dp_{i,k-1}$，那么 $\sum_{k=1}^{i} d_k = dp_{i,i} \le i$，$d_k$ 一共只会有 $\mathcal{O}(\sqrt{n})$ 种值，也就是说这个凸包只有 $\mathcal{O}(\sqrt{n})$ 段。

具体而言，这样求出这 n 个凸包：

- 对于 $k \le \sqrt{n}$ 的部分，直接 dp 出所有 $dp_{i,k}$。
- 对于 $k > \sqrt{n}$ 的部分，凸包的斜率 $x < \sqrt{n}$，那么枚举这个斜率 x，用 wqs 二分的办法（用 $g(l,r) = f(l,r) - x$ 做一遍无限制的区间划分问题得到 $\max(dp_{i,k} - kx)$）得到凸包上这种斜率对应的端点。

那么现在的问题是进行这 $\mathcal{O}(\sqrt{n})$ 次 dp。用线段树是单次 $\mathcal{O}(n \log n)$ 的，无法接受。

考虑 dp 时要干什么：

1. 将 $last_{a_i}$ 到 i 之间的值加上 1。
2. 加入 i 的值。

对于 $j < i$，如果 $v_j < v_i$，那么 j 就没用了。因此我们可以用一个单调递减的单调栈维护有用的位置。
这个单调栈在 1 操作时需要从中间 pop，因此要用链表。

1 操作时需要在栈里面 `lower_bound`，用一个并查集维护。

时间复杂度为$\mathcal{O}(n\sqrt{n}\alpha(n))$，空间复杂度为 $\mathcal{O}(n+m)$。

CF 1943E2 MEX Game 2 (Hard Version)

> 两人博弈。
>
> 初始给定一个数组 a，值域为 $[0,m]$，元素 i 出现了 f_i 次。
>
> A 先手，每次可以选 a 中的某个元素加入集合 c 并删去。
> B 后手，每次可以删去不超过 k 个元素。
>
> 求最优策略下 $\mathrm{mex}(c)$ 的最大值。
>
> $m\le 2\times 10^5, 0\le f_i,k\le 10^9$。

二分答案。

问题转化为了有 x 堆石头，A 每次可以删去一整堆，B 每次可以删去不超过 k 个石子。求 B 能够删掉某一堆石子。

考虑贪心，感受这个选取的过程，注意到几个性质：

- A 每次一定会删去最小的那堆。
- B 获胜的条件是在某个时刻最小的那堆 $\le k$。

这启发我们从小到大排序，但是直接贪心还是不好贪，因为发现对于 B 的策略，肯定是对于那些被删去的堆尽量不去减，但是如果把后面的堆减得过小了又会被 A 删去。但是思考后一种情况，其实不可能发生，因为如果把后面的删的更小了，其实等价于删去前面的那一堆。所以 B 的策略就是在不影响相对顺序的情况下删数。

假设 B 最终赢了，即某一堆 $\le k$，那么 B 显然不会去删比这堆大的堆。考虑直接枚举这一堆的位置 i，那么是从 i 开始一直往前面减，直到相同之后就一起减。这可以 $O(m^2)$ 模拟，分析一下复杂度就是 $O(m^3\log m)$。

考虑优化 check 部分。

思考 check 的过程，相当于不断地把 i 减小到和前面的堆相同大小。每轮下来 A 删开头，然后 B 操作 $[1,i]$ 的一段后缀。起初 A,B 的操作其实是独立的，所以我们可以加速这个过程，直到 A 删到了某个被 B 操作的点。

加速这个过程是简单的，可以直接二分。对于剩下的数，不难发现它们的极差一定 ≤ 1。假设总和为 s，个数为 len。实际上知道了 s,len 就可以知道每一堆的大小。容易发现，A 操作其实等价于 $len\leftarrow len-1,s\leftarrow s-min_v$，B 操作等价于 $s\leftarrow s-K$，同时 $\min_v=\lfloor\frac{s}{len}\rfloor$。

直接设 dp_i 表示长度为 i 的最大的总和使得 Bob 赢：

$$dp_i-\lfloor\frac{dp_i}{i}\rfloor-k\le dp_{i-1}$$

注意到可以把 $\lfloor\frac{dp_i}{i}\rfloor$ 等价成 $\frac{dp_i}{i}$，这只会多减去小数部分，不影响答案，于是有 $dp_i=\lfloor\frac{i(dp_{i-1}+k)}{i-1}\rfloor$。

继续优化，随着 i 的增加，p 肯定是单调的，所以省去了一个二分。

总的时间复杂度为 $\mathcal{O}(m\log m)$。

AT AGC067E Biconnected Graph

> 计算满足以下条件的大小为 n 的图 G 的数量（可以有重边，不能有自环）：
>
> - G 边双连通
> - 删去 G 中任意一条边后，剩下的图不边双连通
>
> $1 \le n \le 50$。

考虑一下删掉一条边会剩下什么东西：
这是一个边双，因此会有一条由若干个边双串起来的链（长度不小于 2），这条边的端点连接链的两端。

那么考虑能不能递归到这些边双内解决子问题。

对于连接链的边，它们的要求已经满足。
小边双内会有一个或两个点被从外部连接，对于小边双内的边，同样考虑将其断掉产生的链，那么要求就是"从外部被连接"的点不属于链上相邻两个边双。

这样我们会得到如下形式的子问题：
计算大小为 n，关键点为 $1 \cdots k$ 的符合条件的边双连通图数，关键点可以从外部相互到达。

计算方式就是选取一条连接 1 的边，其产生的链要求关键点不能处于相邻边双，用背包计算链的数量。

还有一个问题是任选 1 的出边会导致图被计算 $\deg_1$ 次，因此要记录 $\deg_1$。图中可能会出现二重边，二重边只会被计算 1 次而非 2 次，需要单独补回来。

具体地，设 $F_{n,k}$ 表示大小为 n，关键点 $1 \cdots k$ 的图数，$f_{n,k,d}$ 表示 $\deg_1 = d$ 时图的数量，$g_{n,k,0/1}$ 表示当前的链上有 n 个点，k 个关键点，最后一个边双不含/含关键点。
初值 $F_{1,1} = f_{1,1,0} = g_{0,0,1} = 1$，答案 $F_{n,1}$。

转移:

- 对于 $f_{n,k,d}$：
  枚举 1 所在边双大小 n_0，关键点数 k_0，并设 $X = g_{n-n_0,k-k_0,0} \times \binom{n-k}{n_0-k_0}\binom{k-1}{k_0-1}$，有这些贡献：
  - 在 1 上连了两条链上的边：$X \times f_{n_0,k_0,d-2}$。
  - 在 1 上连了一条，$2 \cdots k_0$ 上连另一条：$X \times f_{n_0,k_0,d-1} \times (k_0 - 1)$。
  - 在 1 上连了一条，$k_0 + 1 \cdot n_0$上连另一条：$X \times f_{n_0,k_0+1,d-1} \times (n_0 - k_0)$。
  - 补上二重边的贡献（此时 $k_0 = k$）：$F_{n-n_0,1} \times f_{n_0,k,d-1} \times \binom{n-k}{n-n_0} \times (n - n_0)$。
- 对于 $g_{n,k,0}$：
  枚举当前边双大小 n_0，设 $X = g_{n-n_0,k,0/1} \times \binom{n-k}{n_0}$，有这些贡献：
  - 在同一个点上连两条：$X \times F_{n_0,1} \times n_0$。
  - 在不同点上连：$X \times F_{n_0,2} \times n_0(n_0 - 1)$。
- 对于 $g_{n,k,1}$：
  枚举 n_0, k_0，$X = g_{n-n_0,k-k_0,0} \times \binom{n-k}{n_0-k_0}\binom{k}{k_0}$，有这些贡献：
  - 在 $1 \cdots k_0$ 上连 $1/2$ 条：$X \times F_{n_0,k_0} \times k_0^2$。
  - 在 $1 \cdots k_0$ 上连一条，$k_0 + 1 \cdots n_0$ 上连另一条：$X \times F_{n_0,k_0+1} \times 2k_0(n_0 - k_0)$。

- 在 $k_0 + 1 \cdots n_0$ 上连 $1/2$ 条:
  $X \times F_{n_0,k_0+1} \times (n_0 - k_0) + X \times F_{n_0,k_0+2} \times (n_0 - k_0)(n_0 - k_0 - 1)$。