2025“钉耙编程”中国大学生算法设计暑期联赛（4）命题报告

南京外国语学校

2025 年 7 月 28 日

试题难度

本场比赛的区分度不够，未区分出铜牌和铁牌队伍。抱歉！

预期试题难度：

- Easy：01、09
- Easy-Medium：07、08、12
- Medium：03、04、05
- Medium-Hard：10、11
- Hard：02、06

通过情况

题目编号	[01]	[02]	[03]	[04]
通过提交数	1072	0	20	4
首次通过时间	00:05:18	-	00:39:23	01:26:09
题目编号	[05]	[06]	[07]	[08]
通过提交数	19	0	124	849
首次通过时间	01:51:43	-	01:01:58	00:15:28
题目编号	[09]	[10]	[11]	[12]
通过提交数	965	18	3	115
首次通过时间	00:04:15	01:00:22	04:25:57	01:25:14

1001 电子带负电

由于序列非负，最大值即为序列所有元素之和。

- 研究 l, r 可得它们只被最左最右两端的第一个非零数所限制。设最左边的非零数是 a_x，最右边的是 a_y，则 $l \le x$，$r \ge y$。
- 研究 L, R 可得它们只被最小最大的非零数所限制。设最小的非零数是 X，最大的是 Y，则 $L \le X$，$R \ge Y$。

直接维护。特判全零的答案。
单组数据时间复杂度 $O(n)$。

1009 量子弹弓

【结论 09】：若 p_i 均大于 0 且 $\sum p_i \geq 2n-2$，则答案为 YES，否则为 NO。

基于结论可以快速写出 $O(n)$ 的代码。

1009 量子弹弓中【结论 09】的证明

- 若存在 $p_i = 0$，那么将没有办法离开 i 点，答案为 NO。
- 对于一个合法的排列，所有 $a_i, a_{(i \bmod n)+1}$ 的路径将必然至少经过每条边两次。利用反证法证明：设某条边仅被经过一次，那么但凡从这条边的某一侧进入另一侧后，就没有办法再回到原侧，无法形成点之间的回路。
- 上述结论告诉我们，存在答案的必要条件是 $\sum p_i \geq 2n-2$。接下来证明充分性。

1009 量子弹弓中【结论 09】的证明

- 直接构造。若 p_i 均大于 1，构造一棵菊花即可。
- 否则，将 p 从小到大排序，得到 $p_{x_1}, p_{x_2}, \cdots, p_{x_n}$。题设排列 a 即为此时的排列 x。设满足 $p_{x_i} = 1$ 的 i 的集合是 $\{1, 2, 3, \cdots, k\}$。将 $x_1 \sim x_{k+1}$ 连成一条自上而下的链。设链顶为 x_1，链底为 x_{k+1}。
- 遍历 $j = [k+1, n]$。若 $j \neq n$，向上移动至父亲 x_i 后继续向上移动到 $x_{\max(1, i-p_{x_j}+2)}$，再向下一格构建 $x_{\max(1, i-p_{x_j}+2)}$ 的新儿子 x_{j+1}。这样保证了 $\text{dis}(x_j, x_{j+1}) \le p_{x_j}$。
- 若 $j = n$，移动至父亲 x_i 后直接移动到 $x_{\max(1, i-p_{x_j}+1)}$。这样保证了 $\text{dis}(x_n, x_1) \le p_{x_n}$。
- 我们的目标是从 x_{k+1} 成功移动到 x_1，需要在链上向上移动 k 次。对于 $j = [k+2, n-1]$，我们可以向上移动 $p_{x_j} - 2$，对于 $j = k+1$ 或 $j = n$，可以移动 $p_{x_j} - 1$。

1009 量子弹弓中【结论 09】的证明

- 也就是说，我们只需要满足 $(\sum \max(0, p_i - 2)) + 2 \geq (\sum \max(0, 2 - p_i))$，也就是 $\sum p_i \geq 2n - 2$。这恰是我们初始满足的条件。
- 因此，【结论 09】得到证明。

1007 咖啡的罪恶

【结论 07】：仅有如下几种可行的咖啡序列：

1. 2 0 2 0
2. 1 2 1 0
3. 2 1 2 0 0
4. n-4 2 1 0 0 ... 1 0 0 0

发现所有 $n > 4$ 的咖啡序列不交，所以可以用一个 set 维护所有 $n > 4$ 的咖啡序列，并用线段树维护以每个节点为右端点的咖啡序列长度是多少。每次修改只会影响至多一个 $n > 4$ 的咖啡序列，查询时由于咖啡序列的左端点随右端点递增而递增（咖啡序列不交所以肯定递增），所以直接找到在查询区间内的咖啡序列的右端点区间，查一个区间最值即可。

对于 $n = 4$，暴力维护所有咖啡序列即可。查询时二分。

所以总复杂度是 $O(n \log n)$ 的。精细实现可以跑一秒以内。

1007 咖啡的罪恶中【结论 07】的证明

- 考察去除 a_0 后的数列，所有非 0 数之和为 $n+1-a_0$，而根据 a_0，总共有 $n-a_0$ 个非 0 数。所以这些非 0 数里必定只有一个 2 并且剩下的全部都是 1，因此讨论 2 的位置。此时数列中最多出现 4 种数，进而最多 4 个位置非 0，也就是 $a_0 \geq n-4$。
- $a_2 = 2$，此时 $a_1 = 0/1$，对应 2 0 2 0 和 2 1 2 0 0。
- $a_1 = 2$，此时若 $a_0 = 1$，对应 1 2 1 0，否则对应 n-4 2 1 0 ... 1 0 0。
- 因此，【结论 07】得到证明。

1008 回忆与展望

设答案序列包含 k 个上升子列。为了方便，我们在后 $k-1$ 个子列的最开头暂时添加一个 0，最后去掉。

此时，对于一个数 i，我们希望将它尽量分散着放在所有子列中（因为这可以产生更多的 x 贡献）。设它出现次数为 cnt_i，那么它产生的贡献是 $\max(cnt_i-k,0)y+\min(cnt_i,k)x$。这里由于我们每个子列的最开头都是 0（第一个子列的开头视为 $p_0=0$），所以每个数在它所在的所有子列中都有一个 x 的贡献。

对于 k，总贡献是上述贡献和加上 $(k-1)z$ 减去 $(k-1)x$，加上的数表示两个相邻上升子列的贡献，减去的数表示后 $k-1$ 个子列由于前面有 0，所以让本应该在开头的数移到了第二位，多产生了 x 的贡献。不必在意相邻两个上升子列会不会合并成一个，因为此时统计这个情况时会把一个 x 算成 z，不会导致答案变大。利用桶维护。复杂度 $O(n)$。

1012 海上的太阳

我们发现给定的山坡函数在 $\mathbb{R}_{\geq 0}$ 上是一个下凸函数，同时太阳在第二象限且纵坐标大于所有大楼。于是，大楼底部的横坐标与影子最远处的横坐标呈正相关。

这样一来，我们枚举这些大楼之间的相对顺序，那么最优的建造方案一定是第一座大楼建在原点，接下来的每一座大楼都尽可能靠近原点。可以通过二分等方法求出每一座大楼需要建造在什么位置。

最后需要做的事就是计算从原点走到给定坐标需要的距离。这其实是在计算一段曲线的长度。设曲线的方程是 $y = f(x)$，那么曲线在 $x \in [l, r]$ 的范围内长度就是 $\int_l^r \sqrt{f'(x)^2 + 1}\,\mathrm{d}x$。这个函数不容易直接求原函数，但可以使用辛普森积分的方式近似地求出数值。

1003 维什戴尔

先将 x_i, y_i 对 $sz_{u_i} - x_i, sz_{v_i} - y_i$ 取 max，获得两个连通块中的众数出现次数。

分两种情况讨论：

对于全局有绝对众数的情况：

【结论 03】：绝对众数一定等于 $\min_u\{x_u + y_u\}$。

在得知全局的绝对众数后（设这个绝对众数的值为 0），我们也就知道每棵子树内外的众数分别是多少，判合法性是平凡的树形 dp，如果合法就会产生两个解。

1003 维什戴尔

对于全局 $0, 1$ 数量相等的情况：

首先 n 得是偶数。

显然地，每条边两边众数的值不同，于是我们钦定根为 0，每条边两边 0 出现次数之和必须为 $n/2$，以此可以去掉所有的子树外信息，只需要 dp 出每个点的子树内信息能否达到，这个问题对于每个点做分治 NTT 可以做到三个 log，而对子树大小带权分治可以优化为 $O(n\log^2 n)$。

可能把一些多重背包或者枚举之类的东西拼起来可以优化掉一个 $\log\log n$，不过也无关紧要，两只 log 足以通过。

1003 维什戴尔中【结论 03】的证明

- 若某个叶子上的值为绝对众数的值那么显然正确。
- 否则认为一棵子树是好的当且仅当其子树内的众数不是全局绝对众数，那么由于根是不好的而叶子是好的，一定存在某个不好的点使得其所有儿子都是好的。这个点子树内一定满足不存在绝对众数，也就是说其子树内和子树外的众数分别可以是全局的绝对众数。
- 因此，【结论 03】得到证明。

1004 取石子游戏

【结论 04】：对于一堆包含 x 个石子的堆，若 x 为偶数则 sg 函数值为 0，否则为小于等于 x 的最小质因子的素数的个数。

问题转化为了，ban 掉一些集合，将给定的数不重不漏地划分为若干个集合，记每个集合中元素的最小值组成的集合为 W，要求 W 中的数异或和 $=0$ 和 $\neq 0$ 的方案数，还要满足划分出的每个集合都没有被 ban。

1004 取石子游戏

考虑设计 $dp_{i,S,T}$ 表示划分出的集合中元素最小值只能是前 i 个数，所有数中已经被划分掉的数为 S，划分出的集合中元素最小值为 T 的方案数，S,T 为二进制数。

容易发现 S 的前 i 位必定为 1，而 T 的后 $n-i$ 位必定为 0，所以总计不同的状态只有 $n2^n$ 种，算上暴力转移的复杂度，可以初步分析到 $O(n3^n)$。

但是注意到，实际的子集枚举只需要枚举后 $n-i-1$ 位，所以运算量实际为 $\sum_{i=0}^{n} 3^{n-i-1}2^i = 3^{n-1} \times \frac{1-(\frac{1}{3})^{n+1}}{\frac{1}{3}} = O(3^n)$，并且有大量的状态为 0，复杂度远远跑不满，可以通过。

如果你写了 $O(2^n n^3)$ 的子集卷积，你可能会发现它无法通过这个题，因为容易发现在 $n=18$ 时，$3^n \leq 2^n n^3$。

1004 取石子游戏中【结论 04】的证明

- 考虑归纳，显然基底成立。若对于 $x<k$ 的 x 均满足结论，那么：
- 若 x 为偶数，则 $x-y$ 必然为奇数，sg 值均不为 0，故这一位的 sg 值为 0。
- 若 x 为奇数，设其最小质因子为 a，排名为 r，考察一个素数 $b<a$，容易注意到 $\gcd(x-b,x)=1$，所以所有 $<r$ 的正数均会被取到，同时取 $y=1$，0 也可以被取到。而 r 显然无法被取到，所以这一位的 sg 值为 r。
- 因此，【结论 04】得到证明。

1005 美好的梦境（一）

首先计算出每个 D 在字符串 S（1-idx，长度为 l）中的位置 $p_1, p_2, p_3, \cdots, p_c$。$c$ 是总共 D 的个数。设 $p_{c+1} = l + 1$。

设计 $dp_{i,j}$ 表示此时在第 i 个点，执行子串 $S[p_j, l]$ 后的期望。对于所有 $k \geq j$，预处理执行 $S[p_j, p_k - 1]$（跳过 $p_{j+1}, p_{j+2}, \cdots, p_{k-1}$ 位置的 D）后到达的 $a_{i,x}$ 的 x，记为 $arr_k = x$。

枚举 k 表示执行 $S[p_j, p_k - 1]$ 时 $p_{j+1}, p_{j+2}, \cdots, p_{k-1}$ 位置的 D 在执行时都恰好在一个叶子上执行，统计此时的答案。

1005 美好的梦境（一）

记节点 i 的儿子中，叶子节点数量为 lfs，编号和为 sl，非叶子节点数量为 $sz - lfs$，编号和为 ss。利用这些计算后文的概率。

此时，记 des 表示 $arr_{j+1}, arr_{j+2}, \cdots, arr_{k-1}$ 中不同的数的数量，这表示有 des 个位置的儿子被强制钦定放置叶子。此时钦定 arr_k 的位置放置的不是叶子（导致执行位于 p_k 的 D 的时候会进入这个非叶子节点的儿子）。检查 $des \le lfs$，以及 arr_k 位置是否以前已被钦定放置叶子。去掉这些矛盾点后，利用 des 计算后文的概率。

1005 美好的梦境（一）

对所有非叶子儿子，计算 $\sum dp_{y,k}$， y 为儿子。乘上概率贡献到 $dp_{i,j}$ 中。边界情况是在 $k = c + 1$ 的时候，此时无论到哪个儿子都会直接结算那个儿子的编号。直接计算并乘上概率贡献到 $dp_{i,j}$ 中即可。

复杂度不高于 $O(nl^2)$，可能可以通过严格的分析逼出更低的复杂度上界。但这不在考察范围内。

1010 利物浦集合

题目实际是要求，长度为 n 的，所有元素的范围为 $0 \sim 2^m - 1$，所有元素异或和为 0，任意一个数出现不超过 k 次的可重集的数量。

【结论 10】：异或和非 0 的答案全部相同（即只改变对于异或和的要求，其他要求不变）。

同时我们又容易通过容斥求出总方案数，我们如果能计算出异或和为 0 的方案数和异或和非 0 的方案数的关系（也就是差值），这个题就做完了。

1010 利物浦集合

当 n 是奇数时，每一个异或和为 0 的方案都唯一对应了一个异或和为 $x(1 \le x < 2^m)$ 的方案：我们只需要把所有的数异或 x 即可。所以当 n 是奇数时，答案就是总方案数除以 2^m。

接下来考虑 n 是偶数。

1010 利物浦集合

先考虑 $k=1$ 怎么做。有如下的映射方式：我们将 $(0,2^m-1),(1,2^m-2),\cdots,(2^{m-1}-1,2^{m-1})$ 配对，其中每对两个数的异或和为 2^m-1。我们考虑按照上述列出这些对的顺序（即按照对中较小值排序）依次枚举这些对。

如果一个对中有且仅有一个数在方案中出现，那么我们将这个数异或 2^m-1，就对应了一种异或和为 2^m-1 的方案。对于一个异或和为 2^m-1 的方案，我们采取同样的映射方式，同样可以对应到一种异或和为 0 的方案。

此时，对于所有可以映射出去的异或和为 0 和 2^m-1 的方案，两边构成双射。

1010 利物浦集合

这当然没有做完，毕竟存在一些序列不存在这样的“仅存在一个数”的对。

注意到对于固定的 n，这样的序列的异或和是固定的，当 $\frac{n}{2}$ 为奇数时，这样的序列的异或和为 2^m-1，否则这样的序列的异或和为 0。

同时，这样的序列的个数是好算的，是简单的组合数。

综上所述，我们解决了 $k=1$ 的问题。

1010 利物浦集合

对于 $k \neq 1$，我们使用类似的映射方式。

但是我们发现一个问题，如果只异或一个数，很多情况下会产生歧义（也就是无法一一对应，破坏了双射）。

于是一个新的映射方式出现了：我们找到排序后最靠前的对使得两个位置上的总个数为奇数，并且交换他们。同样容易说明，存在这样的对的序列构成双射。

这样我们只需要考虑剩余的，也就是所有的配对两端的个数和都是偶数的集合就好了。

1010 利物浦集合

考虑使用生成函数技巧。我们记一个配对两边是奇数 + 奇数的生成函数为 F，一个配对两边是偶数 + 偶数的生成函数为 G。那么，我们相当于要求：

$$[x^n]\sum_{i=0}^{2^{m-1}}(-1)^i F^i G^{2^{m-1}-i}\binom{2^{m-1}}{i}$$

使用二项式定理，可以得到上述式子等于：

$$[x^n](G-F)^{2^{m-1}}$$

F, G 都是好求的，所以多项式快速幂容易做到 $O(n\log n)$。

1010 利物浦集合

这个有进一步的优化。我们首先把上述式子列出来：

$$\begin{cases} (1+x^2+x^4+\cdots+x^{2k})^{2^{m-1}} & \text{if } 2 \mid k \\ (1+x^2+x^4+\cdots+x^{k-1}-x^{k+1}-\cdots-x^{2k})^{2^{m-1}} & \text{if } 2 \nmid k \end{cases}$$

我们只关心 n 次项系数，所以对这两个运用容斥原理可以得到以下最终答案：

$$S=\sum_{x=0}^{2^m-1} f(x)=\sum_{i=0}^{\lfloor \frac{n}{k+1} \rfloor} (-1)^i \binom{n-i(k+1)+2^m-1}{2^m-1}\binom{2^m}{i}$$

1010 利物浦集合

$$W_0 = \sum_{i=0}^{\lfloor \frac{\frac{n}{2}}{k+1} \rfloor} (-1)^i \binom{\frac{n}{2} - i(k+1) + 2^{m-1} - 1}{2^{m-1} - 1} \binom{2^{m-1}}{i}$$

$$W_1 = \sum_{i=0}^{\lfloor \frac{\frac{n}{2}}{\frac{k+1}{2}} \rfloor} (-1)^i \binom{\frac{n}{2} - \frac{i(k+1)}{2} + 2^{m-1} - 1}{2^{m-1} - 1} \binom{2^m}{i}$$

$$W = \begin{cases} W_0 & \text{if } 2 \mid k \\ W_1 & \text{if } 2 \nmid k \end{cases}, f(0) = \begin{cases} \frac{S}{2^m} & \text{if } 2 \nmid n \\ \frac{S-W}{2^m} + W & \text{if } 2 \mid n \end{cases}$$

这样我们解决了第一小问。第二小问直接暴力做就行了。因为每一次的复杂度是 $O(\frac{n}{k})$ 的，总的复杂度是 $O(n \log n)$。

1010 利物浦集合中【结论 10】的证明

- 考虑 $f(0)$ 与 $f(x \neq 0)$ 的关系，类似地，我们把异或和为 x 的两个数配对。不难发现，这样的配对和 0 与 $2^m - 1$ 的配对没有本质区别。
- 所以实际上 $\forall x \neq 0, f(0) - f(x)$ 都相同。自然所有的 $f(x \neq 0)$ 都相同。
- 因此，【结论 10】得到证明。

1011 统计连通块

记 $S=1$，$T=2$。以下**连通块**均表示**极大的弱联通子图**。

首先求出图 G 中的强连通分量数目 A 和连通块数目 B。

在图 G 中：

记 S 所在强连通分量中的边为**一类边**；**剩余的边中**，T 所在强连通分量中的边为**二类边**；**剩余的边中**，在 S 到 T 的（不一定要是简单的）路径上的边为**三类边**；**剩余的边中**，属于任意一个强连通分量的边为**四类边**；其余的边为**五类边**。记五种边的数目分别为 cnt_1 至 cnt_5。

先特判 $k=0$。下面对于 S 和 T 在图 G 中的连通性进行分类讨论。

1011 统计连通块

第一种情况：S 和 T 属于同一个强连通分量。

此时，对于图 G^{k-1} 中的每个连通块，这个连通块的任意两点在图 G^k 中都强连通。换句话说，图 G^{k-1} 中的每个连通块，在图 G^k 中都是强连通的。

对于此连通块的每条边，它在被替换为图 G 时，新边中的所有**一类边**将成为新的这个大强连通分量中的边；而**五类边**将独立构成强连通分量。

G^{k-1} 中，每条边将会产生 $A-1$ 个新的强连通分量；而原有的连通块也会成为强连通分量。所以说，G^k 的强连通分量数目，等于 G^{k-1} 中的连通块数目，加上 G^{k-1} 的边数乘以 $A-1$。边数显然是 m^{k-1}，而连通块数目的计算与上面强连通分量数目的计算类似，可以用矩阵或者推式子解决。

1011 统计连通块

第二种情况：S 到 T 的路径与 T 到 S 的路径均不存在。

此时，图 G^{k-1} 中的任意两点在图 G^k 中，一定**不**强连通。那么，这些点在图 G^k 中都可以代表一个强连通分量。

类似情况一，把其余的强连通分量对应到 G^{k-1} 中的每条边上。则 G^k 的强连通分量数目，等于 G^{k-1} 中的**点数**，加上 G^{k-1} 的边数乘以 $A-2$。（注意这里 S 和 T 属于两个不同的强连通分量，所以乘以 $A-2$ 而非 $A-1$。）计算方法同上。

1011 统计连通块

第三种情况：S 到 T 的路径与 T 到 S 的路径，二者恰好有一个存在。

不妨设存在 S 到 T 的路径。还是考虑图 G^{k-1}。对于图 G^{k-1} 中的两个点，若他们强连通，则在 G^k 中依然强连通。记图 G^i 中，**属于任意一个强连通分量**的边数为 X_i，**不属于任何强连通分量**的边数为 Y_i，**强连通分量**数目为 Z_i。

先考虑 X 和 Y 内部的转移。对于图 G^i 中一条**属于任意一个强连通分量**的边，这条边在被替换为图 G 时，对应的 G 中的**一**、**二**、**三**类边将在图 G^{i+1} 中与原来 G^i 中这条边的两个端点同属于一个强连通分量；**五类边**也将成为强连通分量中的边；唯独**四类边**会成为**不属于任何强连通分量**的边。整理一下，X_i 对 X_{i+1} 的转移系数为 $m - cnt_4$；X_i 对 Y_{i+1} 的转移系数为 cnt_4。

1011 统计连通块

类似地，可以推出 Y_i 对 X_{i+1} 的转移系数为 $cnt_1 + cnt_2 + cnt_5$，对 Y_{i+1} 的转移系数为 $cnt_3 + cnt_4$。

根据上面的推导，同样容易得出 X、Y 对 Z 的转移系数，可以用矩阵求出 Z_k。

单组测试时间复杂度 $O(n + m + \log k')$。

1002 Mon2tr

设 n, q 同阶。以下点的编号均用 i 表示。

首先我们考虑树剖线段树。

定义一个点的附属子树是这个点的子树去掉它重儿子的子树。

每条重链先从 x 跳到这条重链上的位置（记这个位置为 x'）切成两段（对于 x'，我们特殊处理一下，对其所有轻儿子建带权平衡树套树，线段树里维护 i 的信息），每段在线段树上切成一些结点。现在我们有 $O(\log n)$ 个结点，对于每个结点区间，这个区间内的所有点的附属子树要么 v 均为 x'，要么一个点的附属子树的 v 都是这个点本身。

那我们对于线段树使用树套树，每个结点去维护一个数据结构解决这两类问题即可。

1002 Mon2tr

我们首先解决 $v_i = x'$ 的情况。此时除了 i 都是已知的，则我们要求在一个 dfs 序区间内的 i 的最大值。离散化后维护一个全局的主席树，在其上二分即可，这部分整体空间复杂度 $O(n\log n)$，单次查询复杂度 $O(\log n)$。

对于另一种情况，相当于我们有一些从开始就确定，不会更改的 (i, v_i)，然后求题意中的式子。

这里可以开一棵持久化线段树，然后按照 $i + v_i$ 从小到大排序，依次做区间 chkmax，这个区间是 $[i - v_i, i]$。查询就在 z 对应的版本查 y 的值。

复杂度 $O(\log^3 n)$，使用全局平衡二叉树技巧即可做到时空 $O(n\log^2 n)$。此时空间不满足要求。

1002 Mon2tr

考虑进一步优化空间，这里唯一困难的地方只剩下给定一些 (i, v_i)，每次查询一些东西了。这里称这些线段树上的结点对应区间为第一类区间。

我们可以发现，题目中的限制实际上是要求了三件事情：

1. $i \geq y$
2. $i - v_i \leq y$
3. $i + v_i \leq z$

每条限制可以在平面上画出一个直线。三条直线围成了一个等腰直角三角形，我们要求这个等腰直角三角形内的最大的 $i + v_i$。我们画出这个等腰直角三角形的斜边上的高（也就是 $v_i = (z - y)/2$），然后我们把之前在树剖线段树上表示的那些结点，进一步按照这个直线划分。这里显然所有第一类区间的深度是无交的。

1002 Mon2tr

现在所有第一类区间一定都在这个直线的一侧。我们发现一个惊人的事实:

v_i 小的这一侧，只需要满足 1. 2. 两条直线，由于 $v_i < (z-y)/2$ 自动满足第三条直线的限制。

同理，另一侧只需要满足 1. 3. 这两条的直线的限制。

现在我们相当于减少了一个自由度，问题变得好解决起来。在使用全局平衡二叉树，并对每个节点中所维护的数离散化之后，我们相当于要对于长度为 $O(n\log n)$ 的序列，求区间中这两种限制下 $i+v_i$ 的最大值。

1002 Mon2tr

对于第一种情况，条件等价于 $i - v_i \le y \le i$，相当于区间 chkmax 单点查询。

对于第二种情况，我们离散化后以 $i + v_i$ 为下标建线段树，节点处维护 i 的最大值，在线段树上二分即可。

使用全局平衡二叉树后，整体的空间复杂度优化为 $O(n\log n)$，时间则保持 $O(n\log^2 n)$。

1006 美好的梦境（二）

设 n,q 同阶。考虑树形 dp。记左右下箭头为 LRD。

设计三元组 $f_{i,il,ir,id}$ 表示在到达节点 i 且曾经是/否到达过最左侧、最右侧、叶子节点的前提下，用掉了至少多少个 LRD。容易使用 dfs 更新所有状态。

这是基于【结论 06】：对于一个状态 i,il,ir,id，不可能出现 R 多 L 少、或者 R 少 L 多的情况。存在一个三元组偏序剩余的三元组。

接下来，暴力做法是：从大到小枚举编号，判断在是/否到达过最左侧、最右侧、叶子节点的前提下能否可以用光所有箭头即可。

通过在最左侧节点使用 L、最右侧节点使用 R、叶子节点使用 D 可以处理多余的箭头，使用 LR 或 RL 可以同时处理左右箭头各一个。此时时间复杂度 $O(n^2)$，无法通过。

1006 美好的梦境（二）

注意到对于一组 $\{il, ir, id\}$ 是一个关于 l, r, d 的三维偏序。具体而言（设 l', r', d' 为至少用掉的 LRD 的数量）:

- $d - d' \geq 0$ 或 $d - d' = 0$（根据 id 的值讨论）
- $l - l' \geq 0$
- $r - r' \geq 0$
- $l - l' \leq r - r'$，$l - l' \geq r - r'$，$l - l' = r - r'$（根据 il, ir 的值讨论，通过移项等方法合并偏序关系）

  对一共 8 种情况做二、三维偏序即可。
  时间复杂度 $O(n \log^2 n)$。**代码量较大**。

1006 美好的梦境（二）中【结论 06】的证明

- 考虑从起点到终点的唯一简单移动路径。
- 此时，要使得碰到最左最右最下，必然要使得 LR 的数量同时增加一个相等值。
- 而 D 的数量是定值，为起点终点深度差。
- 因此，取最优的情况即可。
- 因此，【结论 06】得到证明。