第 49 届 ICPC 国际大学生程序设计竞赛邀请赛武汉站

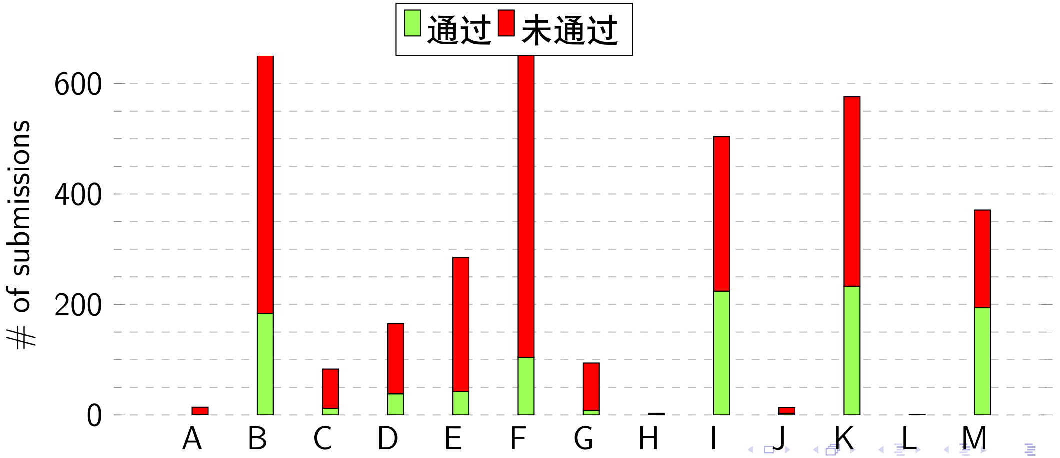
WHU ICPC 集训队

May 2nd, 2024

比赛小结

- 本次比赛共收到 3106 份提交代码。
- 其中 890 份代码正确。
- 224 个队伍有提交记录。
- 224 个队伍至少通过一题。
- 特别地，ChatGLM 也参与了这次比赛，但是未能通过题目

各题通过情况



I. Cyclic Apple Strings

题意

给定一个长度为 n 的仅由 0 和 1 组成的字符串 s。
可以进行如下操作任意次：选择 s 的一个子串和一个正整数 k，并将子串向左循环移动 k 位。求出将 s 变为有序的所需的最小操作次数。

I. Cyclic Apple Strings

题解

观察到每次操作只能减少一个不联通最右侧的连续的 1 块，故答案是所有不联通右侧的 1 块的数量，暴力数出来即可。时间复杂度 $O(n)$。

K. Party Games

题意

有 n 个整数 $1, 2, 3, \ldots, n$ 从左到右顺序排成一行，每轮可以从最左端或最右端移走一个数。若移走前，数已经拿完或剩下数异或和为 0，当前行动者负，问先手是否必胜。

K. Party Games

解法

回顾 $\oplus_{i=1}^{n} i$ 的值，有：

$$\oplus_{i=1}^{n} i = \begin{cases} 1 & n \bmod 4 = 1 \\ n+1 & n \bmod 4 = 2 \\ 0 & n \bmod 4 = 3 \\ n & n \bmod 4 = 0 \end{cases}$$

容易发现，$n \bmod 4 = 0, 1$ 时，先手可以先拿走最左或最右的数令异或和为 0，因此先手必胜，$n \bmod 4 = 3$ 时，先手无法操作，负。$n \bmod 4 = 2$ 时，先手无论拿最左侧还是最右侧，对手总能拿走另一侧使得异或和为 0，因此先手必负。

B. Countless Me

题意

给定一个长度为 n 的非负整数数组 a_i，可以进行如下操作至多 n 次：选择 i, j, x，令 $a_i \leftarrow a_i + x$，$a_j \leftarrow a_j - x$。
最小化操作后 $a_1|a_2|\dots|a_n$ 的值。

B. Countless Me

题解

注意到给出的操作次数可以做到任意分配，我们算出数组 a 所有元素的和 sum，可以将题意转化成把 sum 分配给 n 个位置，使得所有位置的数的或最小。

考虑从最高位向最低位贪心，假设当前位是第 bit 位，显然如果当前位不放数字时仍能够将剩余的 sum 分配给后续位，则显然不在这一位放数字。否则，考虑在这一位放置多少数字合适，令最多能在这一位放置 x 个数，显然 $x = \min(n, \lfloor \frac{sum}{2^{bit}} \rfloor)$ 如果当前放置的数不到 x，则后续其余位的和大于 2^{bit}，由于满足进位关系，容易证明出存在一个相同结果或者更好结果的方案使得当前位多放置一个，故可以证明出最优解中如果当前位要放置就放到 x 个。按照这个方案模拟即可。

F. Custom-Made Clothes

题意

交互题。有一个 $n \times n$ 方阵 $a_{i,j}$，满足 $a_{i,j} \geqslant a_{i-1,j}, a_{i,j} \geqslant a_{i,j-1}$。
每次可以询问一个位置是否小于等于某一个值，要求 50000 次询问内得到方阵 k 大值。
$n \leqslant 1000, a_{i,j} \leqslant n \times n$。

F. Custom-Made Clothes

解法

二分答案。考虑当前二分了值 x，现在通过询问找到方阵有多少个值小于等于其。

从第 n 行第 1 列开始询问，假设目前在 r 行 c 列。若当前位置值小于等于 x，说明在第 c 列上，r 行之前的所有行在这一列的值都小于等于 x，移动到 r 行 $c+1$ 列重复询问；若当前位置值大于 x，移动到 $r-1$ 行 c 列重复询问。注意走到边界时仅会产生一种移动。

发现上述过程实际上是从 $(n,1)$ 只能向上或右走到 $(1,n)$ 的一条路径，长度不超过 $2n$，加上二分答案，总次数不超过 $2n\log n$。

E. Boomerang

题意

给定一个 n 个节点的树，根节点为 r，设 $V(r,t)=\{v|\text{dis}(r,v)\leqslant t\}$，其中 $\text{dis}(u,v)$ 表示树上两点 u 和 v 间的唯一简单路径的边数。再给定一个 t_0，对每个 k，你需要选择一个点 r_0，定义 $V'(r_0,t)=\{v|\text{dis}(r_0,v)\leqslant k(t-t_0)\}$。找到一个最小的 t，使得 $V(r,t)\subseteq V'(r_0,t)$。

E. Boomerang

题解

注意到，对于每个可能是答案的 t，选择子树 $V(r,t)$ 的直径中点作为 r' 一定是最优的。
因此考虑维护每个时刻的树直径。可以观察到，如果当前树的直径端点是 (u,v)，新加入一个节点 x 后，新直径端点一定是 $(u,v),(u,x),(v,x)$ 三者其一，我们只需要维护任意一个可以处理树上节点之间距离的算法即可。
维护树直径之后，观察到 k 变大时，t 不会变大，因此可以用双指针处理出答案。维护直径复杂度 $O(n\log n)$，统计答案复杂度 $O(n)$。

D. ICPC

题意

在 n 个排成一列的格子中，第 i 格子上都有一个数 a_i。对所有 $1 \leqslant s \leqslant n, 1 \leqslant t \leqslant 2n$ 求出从第 s 个格子出发，每次移动至任意一个相邻的格子，移动 t 次后到达过的格子上数的和的最大值。

题解

本题允许复杂度为 $O(n^2)$ 的解法通过。
不难发现对于任意 s, t，存在至少一种最优走法仅改变移动方向至多一次。

D. ICPC

题解

设 $g_{i,j}$ 表示从 i 出发移动不超过 j 次，且不改变移动方向时经过格子上数的和的最大值。g 可以通过前缀和快速求出。
对于不改变移动方向的走法，$g_{s,t}$ 即为答案。
对于改变移动方向恰好一次的走法，若其在 x 折返，$g_{x,t-|t-s|}$ 即为答案。
故我们只需求
$F_{s,t} = \max\{g_{s,t}, g_{s+1,t-1}, g_{s+2,t-2}, \ldots, g_{s-1,t-1}, g_{s-2,t-2}, \ldots\}$。这是一个前后缀最大值形式的递推，在预处理出 g 后可以 $O(n^2)$ 得到 F。
还有许多其他的 $O(n^2)$ 做法，在此不做赘述。

M. Merge

题意

给定由 n 个数组成的序列，每次操作可以合并大小相差为 1 的数字并得到他们的和，求任意次操作后剩下的数的最大字典序

解法

注意到每次操作都需要一个偶数和一个奇数并产生一个奇数，故可能合并产生的最大数为 $2x+1$ 或 $2x-1$，其中 x 为当前序列中的最大偶数。我们只需要不断判断是否能够得到 $2x+1$ 或 $2x-1$，并将相应数合并即可。

M. Merge

解法

可以通过如下方法判断是否可以得到某个数 v:

- 若当前序列中有 v，则可以得到 v
- 若当前序列中无 v 且 v 为奇数，则仅在能得到 $\lfloor \frac{v}{2} \rfloor$ 和 $v - \lfloor \frac{v}{2} \rfloor$ 时可以得到 v
- 若当前序列中无 v 且 v 为偶数，则不能得到 v

该过程可以通过递归实现，且判断 v 是否可以得到仅需要递归 $O(\log v)$ 次。

若我们确定 $2x \pm 1$ 可以得到，仿照判定的过程将相应的数合并为 $2x \pm 1$ 即可。可以证明这样能达成最大的字典序。

如果使用哈希表维护各个数的出现次数，可以在 $O(n \log V)$ 时间内解决本题。

C. TreeBag and LIS

题意

给定一个整数 x，构造一个仅由数字字符 $0, 1, 2, \ldots, 8, 9$ 构成的长度不超过 10^5 的字符串 s，
使 s 的所有最长上升子序列的值之和为 x。
$x \leqslant 10^{13}$。

C. TreeBag and LIS

解法

这是一道十分开放性的题目，几乎每份通过代码做法都不同。以下题解来自出题人的做法：
一个容易想到的做法是使用 01 构造：
考虑在字符串 0111...111 的不同位置插入 0 调整答案。
在每个位置插入 0 的贡献为该位置后续 1 的个数。
此做法需要 $2\sqrt{x}$ 的长度，不足以通过本题。

解法

通过在不同的位置插入 0 来调整答案看上去是很不错的思路。
那么怎么优化长度呢？

C. TreeBag and LIS

解法

对于 x 较小的情况，使用上述 01 构造即可。
下面考虑 x 足够大的情况：
很容易想到，使用更长的 LIS 构造可以让字符串的值显著增大。
同时，插入 0 的位置越靠左，对字符串值的贡献越大。
考虑类似进制的思路，贪心的从左到右考虑每一个位置插入 0 的个数。
发现只需要保证在最左边插入 0 的贡献足够大，
并且相邻两个位置插入 0 的贡献比值较小，
即可保证插入 0 的总数不会过多。
考虑取 LIS 长度为 3 构造：
一个可行的构造是 8...89...97...78...86...67...75645342312.
设置合理的 6, 7, 8, 9 的个数即可满足上述条件。

C. TreeBag and LIS

解法

一个可行的构造是 8...89...97...78...86...67...75645342312.
此时会发现，我们插入一个 0 最少会获得 12 的贡献。
因此纯粹贪心的插入可能导致最后剩余一个小于 12 的贡献无法填补。
考虑在 2312 前插入 0 的贡献是 $12 + 23 = 35$，
注意到在 $12x + 35y = n$ 对任意的 $n > 12 * 35$ 有解。
因此留出 $12 * 35$ 以上的贡献，
使用最后两个插入位置用 $exgcd$ 求解填补答案即可。
该做法需要 $O(x^{\frac{1}{3}})$ 的长度，可以通过本题。
另外，本题长度限制较为宽松，
通过设置更长的 LIS 长度，可以用更少的长度完成本题。

G. Pack

题意

给定 n 件 A 物品和 m 件 B 物品。每件 A 物品的价值均为 a，每件 B 物品的价值均为 b。你现在需要确定一种商品的包装方案，包含若干件 A 物品和若干件 B 物品，满足物品的价值和为 k。包装出尽可能多的商品，直到剩下的 A、B 物品不足以包装成一件商品。请你求出在所有可能的包装方案中，A、B 物品剩余数量和最小值。

G. Pack

题解

假设一种包装方案为 x 个 A 物品和 y 个 B 物品，那么能组成的包装数量为 $\min(\lfloor\frac{n}{x}\rfloor, \lfloor\frac{m}{y}\rfloor)$，要求的答案为 $(x+y)\times\min(\lfloor\frac{n}{x}\rfloor, \lfloor\frac{m}{y}\rfloor)$。

对于 $a\times x+b\times y=k$ 这一限制，可以使用 exgcd 求出 x, y 的最小变化量 $\Delta x, \Delta y$。

一个简单的暴力做法就是不断地对 x, y 进行最小变化 $(x+\Delta x, y+\Delta y)$，找出使 $(x+y)\times\min(\lfloor\frac{n}{x}\rfloor, \lfloor\frac{m}{y}\rfloor)$ 最大的 x, y。

考虑优化这个过程，发现 $\min(\lfloor\frac{n}{x}\rfloor, \lfloor\frac{m}{y}\rfloor)$ 这一部分是可以用除法分块来处理的，那么对于根号个 $\min(\lfloor\frac{n}{x}\rfloor, \lfloor\frac{m}{y}\rfloor)$ 值相同的 x, y 区间，只要找到最大的 $x+y$ 即可。时间复杂度 $O(\sqrt{n})$。

J. Gensokyo Autobahn

题意

给定一张有向图和序列 $a_1, a_2, \ldots, a_k$。对 $i = 1 \ldots k$ 执行：随机选择有向图中原本存在的一条边，加入 a_i 条该边的拷贝。求最终有向图中从 1 到 n 的最短路个数的期望。

解法

对每一条原图中的最短路，考虑其在加边后所有边重数（即该边被修建的条数）的乘积的期望 E。不难发现 E 对原图中所有最短路都是相同的，故最终答案为 $E \times c$，其中 c 为原图中最短路个数。
于是只需计算 E。E 可以通过动态规划在 $O(mk)$ 时间内求出，该做法不能通过本题，在此不做赘述。接下来介绍一个 $O(k \log^2 k)$ 的多项式做法。

解法

将第 i 次加入的边的颜色视为 i。那么对于所有包含且仅包含集合 S 中颜色的边的最短路，其对 m^kE 的贡献为 $m^{k-|S|}\binom{d}{|S|}|S|!\prod_{i\in S}a_i$，其中 d 为最短路长度。

于是

$$
\begin{aligned}
m^kE &= \sum_S m^{k-|S|}\binom{d}{|S|}|S|!\prod_{i\in S}a_i \\
&= \sum_x m^{k-x}\binom{d}{x}x!\sum_{|S|=x}\prod_{i\in S}a_i
\end{aligned}
$$

解法

该式在求出 $\sum\limits_{|S|=x}\prod\limits_{i\in S}a_i$ 后可线性计算。而 $f_x=\sum\limits_{|S|=x}\prod\limits_{i\in S}a_i$ 实质是一个背包，其普通型生成函数为 $\sum\limits_{i=0}^{k}f_iz^i=\prod\limits_{i=1}^{k}(1+a_iz)$，可利用多项式分治乘在 $O(k\log^2 k)$ 时间内算出。

A. Shaking Trees

题意

有一棵包含 n 个节点的有根树，节点编号为 1 至 n。树的根为节点 1，且**树的高度不超过** 100。每次操作时可以选择一个节点 u，使节点 u 与其父节点断开 (如果有)，成为一颗新树的根节点，然后删除以节点 u 为根的树中的所有叶节点。

求删除所有节点所需的最少操作次数和通过最少次操作删除所有节点的方案数，对 10^9+7 取模。

同时保证 $n \leqslant 2\times10^5$，树的深度不超过 100。

A. Shaking Trees

题解

首先我们发现一直选根节点一定是最优的，能达到最小的操作次数，即树的深度 d（假设根节点的深度为 1)。
这是因为：最终答案一定不会比深度 d 小。因为每一次操作最多可以使这颗树的深度（如果变成了两棵树，则是两棵树的深度和）减 1。换句话说，你无法通过一次操作使得深度减 2 或更多。同时，一直选根节点，你就只需要恰好 d 次操作，因为每次都能让深度减 1。
因此，第一问的答案，删除所有节点所需的最少操作次数就是 d。

A. Shaking Trees

题解

同时，我们还可以发现，实际上只要保证：每次选的点都在由最深的叶节点到根节点的路径上，那么就一定能保证每次操作也能让树的深度减 1。
因此，实际上我们的选点只会出现在一条链上。所以我们有了很朴素的在这条链上 dp 的想法。
但是又有许多问题。比如这条链上不是所有点都能选。可以发现，对于这棵树上的点 u，只有当没有其他点和点 u 深度相同时，我们才能选点 u。
如果同时有多个叶节点深度都最大，那么就必须要在这几条路径的重合路径上。
并且在切断之后，也会让有些点变得可选。所以我们需要在 dp 的时候解决这些问题。

A. Shaking Trees

题解

首先我们先把最长链从树里面取出来，假设其长度为 d。因为是在若干个最长链的交上，所以任选一条链都可以。在这条链上进行区间 dp。
对于闭区间 $[l, r]$，l 是深度大的那端，r 是深度小的那端，则说明对 $l-1$ 和 r 分别操作了一次，且中间 $[l, r-1]$ 都没有被选中过。此时枚举区间中的点 i，表示对点 i 操作一次，将 i 和 $i+1$ 断开。但是有些点是可以选的，有些点是不能选的。

A. Shaking Trees

题解

怎么看能不能选呢？从深度小的往深度大的枚举，深度最小的点 (其实就是 r) 肯定可选，然后再维护深度比点 i 大的点的其他分支，拉下来最深的有没有超过 i。其实就是看祖先从不同儿子获得的次深链，减去前面选过祖先的次数，是不是比 $dep[r-1]$ 小，如果小那就可以，否则就不行。所以我们发现，我们还需要一维表示在现在这个状态之前，选了祖先多少次。实际上祖先可能选了 $0 \sim (d-r+l-1)$ 次。

A. Shaking Trees

题解

因此我们得到了最终的 dp 方程：
l,r 表示现在正在处理闭区间 $[l, r]$。l 是深度大的那端，r 是深度小的那端。t 表示此时选了 t 个比 r 的深度还小的。枚举 i 表示现在打算选点 i，把点 i 和点 $i+1$ 之间断开。
所以 $f[l][r][t]+=f[l+1][i][t+1]*f[i+1][r][t]$
但是因为组内有序，组间无序，所以还是要乘一个 $C(r-l, i-l)$
所以最后是 $f[l][r][t]+=f[l+1][i][t+1]*f[i+1][r][t]\times C(r-l, i-l)$
因此这题最终能在 $O(N+d^4)$ 时间复杂度内被解决。题中限定 $d \leqslant 100$，实际上 $O(d^4)$ 的区间 dp 很难跑满，正常写法完全能在 $1s$ 内通过此题。而暴力剪枝做法也基本上被特殊数据卡掉无法通过。

H. Wings of Crystals

题意

给定一棵带点权的树。要求将树的某些边断开，形成若干条链。链的权为其包含的所有点的权的和，求所有链的权的平方的和的最大值。

H. Wings of Crystals

题意

给定一棵带点权的树。要求将树的某些边断开，形成若干条链。链的权为其包含的所有点的权的和，求所有链的权的平方的和的最大值。

题解

本题标程复杂度为 $O(n\log^2 n)$。
不难得到动态规划做法：$dp_{i,j}$ 表示以 i 为根的子树向上贡献权为 j 的链时子树内链的权的平方的和的最大值，由于 j 仅有 $O(\mathrm{size}_i)$ (size_i 代表子树大小) 种取值，由树形 DP 可 $O(n^2)$ 求解，但不足以通过本题。

H. Wings of Crystals

题解

观察到树形 DP 复杂度的瓶颈是合并两个兄弟的 dp 的过程。
考虑两个兄弟 x, y 合并的过程。对于所有可能的 j，我们要找出相应的 k，使得 $dp_{x,j} + dp_{y,k} + (j+k)^2$ 最大。对式子稍做处理，可得 $dp_{x,j} + j^2 + (dp_{y,k} + k^2 + 2jk)$，其中对于确定的 j，$dp_{x,j} + j^2$ 为常数，我们只需最大化 $dp_{y,k} + k^2 + 2jk$。不难发现其为一条斜率为 $2k$，截距为 $dp_{y,k} + k^2$ 的直线在 $x = j$ 处的值。

H. Wings of Crystals

题解

于是该式最大值可以用李超树/单调栈维护 y 的 dp 代表的直线求出；设我们已从儿子 y 处继承得到其 dp 构成的李超树/单调栈，那么将儿子 x 中的每个 dp 逐个并入 y 的复杂度为 $O(\text{size}_x \log \text{size}_y)$。
于是我们可以启发式合并所有儿子，以 $O(n \log^2 n)$ 的复杂度解出该题。

L. Magic Fairies

题意

给 n 个高度互不相同的在数轴上紧挨着的柱子，q 次独立询问从第 x 个柱子上降下 V 单位的水，按真实重力流动完毕后有多少个柱子被水覆盖。

L. Magic Fairies

解法

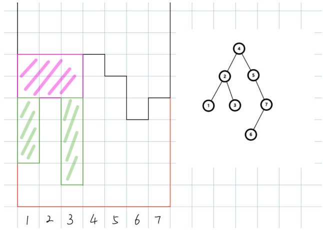
本题解法为在线的 $O(n \log n)$，但是允许可能的 $\log^2$ 做法，即仅需序列二分的做法。
首先对柱子 (i, h_i) 按照下标 i 为二叉排序树关键字，h_i 为大根堆关键字建立笛卡尔树。

L. Magic Fairies

题解

每个节点 u 需要维护子树大小，向上的倍增数组和 vr_u, vl_u，$sum_u = vl_u + vr_u$。其中 vr_u 表示水平面从 h_u 位置到达 h_{par_u} 位置需要的水量，vl_u 表示淹没子树 u 直到 h_u 水面需要的水量。其中，par_u 表示 u 在笛卡尔树上的父亲。

以右图为例子，$h = [2, 5, 1, 7, 6, 4, 5]$ 建出笛卡尔树，当 $u = 2$ 时，绿色部分为 vl_2 的水量，紫色部分为 vr_2 的水量。

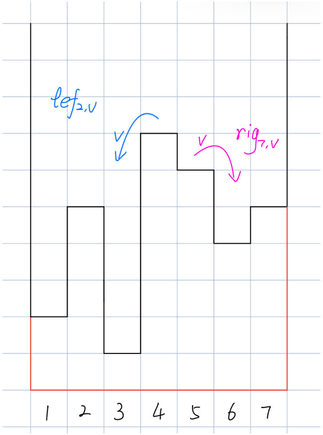


L. Magic Fairies

题解

在回答询问之前，先定义子问题 $lef_{x,V}$ 表示从子树 x 中根左右 dfs 序最大的点垂直降下 V 的水，且不会淹没到 x 父亲的最终覆盖柱子的个数，其本质是从 x 的父亲 p 流过来的水造成的影响，且 x 是 p 的左孩子，类似的，定义 $rig_{x,V}$ 表示从 dfs 序列最小的点降水，即 x 是右孩子。

例如右图，$lef_{2,V}$ 是从第 3 个柱子降水，也是柱子 4 流向左侧的水，$rig_{7,V}$ 是从第 6 个柱子降水，也是柱子 5 流向右侧的水。



L. Magic Fairies

解法

对于询问 (x, V):

- 若 $V \leqslant sum_x$:
  - 若 $vl_x < V$，答案为 siz_x。
  - 若 $vl_x \geqslant V$，此时若两个孩子 ls, rs 不会全被淹没，则答案为 $lef_{ls,\frac{V}{2}} + rig_{rs,\frac{V}{2}}$，否则按容量重新分配一下水量。
- 若 $sum_x < V$ 则首先通过倍增寻找最深的 y 满足 $V \leqslant sum_y$。
  - 若 $vl_y < V$，答案为 siz_y。
  - 若 x 在 y 右子树，答案为 $lef_{ls_y,V-sum_{rs_y}} + siz_{rs_y}$。
  - 若 x 在 y 左子树，答案为 $rig_{rs_y,V-sum_{ls_y}} + siz_{ls_y}$。

L. Magic Fairies

题解

这里仅说明 $lef_{x,V}$ 的情况，rig 的情况类似。考虑某子树 $h = [4, 7, 1, 5, 3, 8, 2, 9, 6]$，需要回答 $lef_{8,V}$ 即从右侧父亲降水，观察可得，水覆盖的顺序为 $9, 7, 5, 3, 4, 1, 2, 6, 8$，是右左根的 dfs 序，且覆盖 v 节点使用的水量为 vr_v。
那么可以预处理好右左根 dfs 序上，vr_v 的前缀和，那么在这个前缀和上二分即可找到固定水量 V 能够覆盖这棵子树的具体节点个数。