信息学竞赛
数学基础

主讲：李宁远

1 组合数学
- 组合数学基础
- 组合数的计算
- 卡特兰数

2 数论
- 数论基础
- 模意义下的运算体系
- 一些有用的定理
- End

组合数学基础

- 组合数学基础

一些恒等式

- 组合数的对称性：$\binom{n}{m}=\binom{n}{n-m}$。
- 二项式定理：$(x+y)^n=\sum_{i=1}^{n}\binom{n}{i}x^iy^{n-i}$。
- 范德蒙德卷积：$\sum_{i=0}^{k}\binom{n}{i}\binom{m}{k-i}=\binom{n+m}{k}$，OI-Wiki。
- 一行组合数中奇数位置和偶数位置的和：$\sum_{i\quad,i\le n}\binom{n}{i}=\sum_{i\quad,i\le n}\binom{n}{i}=2^{n-1}$。证明可以考虑对于前 $n-1$ 个任意一种选法，都可以调整最后一个选不选使得选了奇数个或选了偶数个。

组合数前缀和

纵向前缀和：$\sum_{n=k}^{m}\binom{n}{k}=\binom{m+1}{k+1}$，可以由递推式推导。正斜向的前缀和可以转化为纵向的。
多次询问组合数的横向前缀和做法十分复杂，网上有博客，可以自行搜索。有莫队做法和双 log 做法。一般说组合数前缀和也是指横向。

例题 1

洛谷 P2181 对角线

例题 2

洛谷 P8106 数学练习

容斥原理

OI-Wiki

例题

洛谷 P1450 [HAOI2008] 硬币购物

组合数的计算

一般而言，由于组合数增长较快，都会在对某个模数取模的情况下计算组合数。

平方递推

$\binom{n}{m} = \binom{n-1}{m-1} + \binom{n-1}{m}$

组合意义证明：考虑 n 个元素的最后一个选不选。

该方法不要求模数是质数，可以 $O(nm)$ 预处理，$O(1)$ 查询。

平方递推

```
const int mod = 1e9 + 7, N = 5005;
int C[N][N];
void init() {
    for (int i = 0; i < N; i++) {
        C[i][0] = C[i][i] = 1;
        for (int j = 1; j < i; j++)
        C[i][j] = (C[i - 1][j] + C[i - 1][j - 1]) % mod;
    }
}
// C[n][m] 就是组合数
```

例题

洛谷 P2822 组合数问题

模数是大质数时的 O(n) 预处理

由于模数是大质数（大于 n），所以可以认为在 n 以下的数都有逆元。于是预处理阶乘和阶乘逆元即可 $O(1)$ 计算。有一个一次循环处理阶乘、阶乘逆元、单个数的逆元的写法（也是我一般的写法），见下。原理可以看 OI-Wiki。

代码

```
ll f[N], fv[N], iv[N];
void init() {
    f[0] = fv[0] = iv[0] = f[1] = fv[1] = iv[1] = 1;
    for (int i = 2; i < N; ++i)
    f[i] = f[i - 1] * i % mod,
    iv[i] = (mod - mod / i) * iv[mod % i] % mod,
    fv[i] = fv[i - 1] * iv[i] % mod;
}
ll C(int n, int m) {
    return n < m || n < 0 || m < 0 ? 0 :
    f[n] * fv[m] % mod * fv[n - m] % mod;
}
```

模板题

洛谷 B3717 组合数问题

当 m 很小的时候计算单个组合数

当 m 很小时，可以直接计算 $\frac{n(n-1)\cdots(n-m+1)}{m!}$。

- 如果模数是质数，可以计算 $m!$ 的逆元，然后乘上去，复杂度 $O(m)$。
- 如果模数是合数，就需要枚举 $1 \sim m$ 的每个质因子，计算其在 $m!$ 中的出现次数，然后把在 $n, n-1, \cdots, n-m+1$ 中的部分除掉，复杂度 $O(m \log \log m)$。

某个质数 p 在 $n!$ 的唯一分解中的出现次数是 $\sum_{i=1}^{\infty} \lfloor \frac{n}{p^i} \rfloor$。

合数模数下的计算

```
ll C(ll n, int m) {
    vector<ll> a(m);
    ll k = n - m + 1;
    for (ll i = 0; i < m; ++i) a[i] = i + k;
    for (int p : prime) {
        if (p > m) break;
        ll vp = 0, pq = p;
        while (pq <= m) vp += m / pq, pq *= p;
        for (ll t = n / p * p; t >= n - m + 1; t -= p)
        while (vp && a[t - k] % p == 0)
        a[t - k] /= p, --vp;
    }
    ll ans = 1;
    for (ll i : a) ans = ans * (i % mod) % mod;
    return ans;
}
```

模小数时的 $O(\mathrm{mod})$ 算法

Lucas 定理 用于解决模数为质数时的问题，见下面数论章节。

例题

洛谷 P5481 [BJOI2015] 糖果

引入

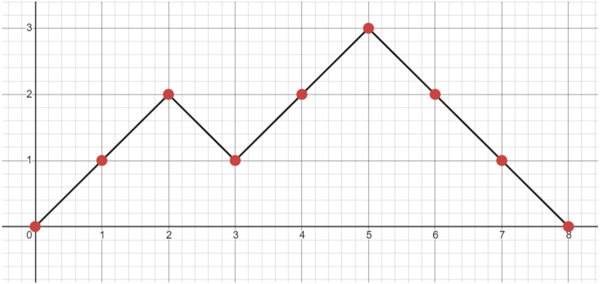
我们来思考这样一个问题:

- 若一个序列由括号组成，且序列中的括号成对出现，则我们称之为一个合法的括号序列，简称为括号序列。
- 求长度为 n 的括号序列的种数。

在我们解决这个问题前，我们来看一些与这个问题等价的问题模型。

路径模型 1

我们定义从 $(0,0)$ 到 $(n,0)$ 的合法路径为每一次仅能从 (x,y) 移动到 $(x+1,y+1)$ 或从 (x,y) 移动到 $(x+1,y-1)$ 且 y 值永远为非负整数。下图则为一条从 $(0,0)$ 到 $(8,0)$ 的合法路径。

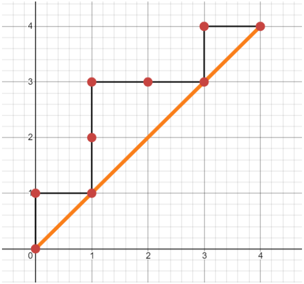


路径模型 1

我们可以将每一次从 (x, y) 移动到 $(x+1, y+1)$ 转化为一个左括号，将从 (x, y) 移动到 $(x+1, y-1)$ 转化为一个右括号，则每一条从 $(0, 0)$ 到 $(n, 0)$ 的合法路径都可以唯一对应一个长度为 n 的括号序列。
上图就转化为括号序列就是 (() (()))。

路径模型 2

我们定义从 $(0,0)$ 到 (n,n) 的合法路径为每一次仅能从 (x,y) 移动到 $(x,y+1)$ 或从 (x,y) 移动到 $(x+1,y)$ 且 $x \leq y$ 永远成立，也可以理解为永远在 $y=x$ 的左上方。下图黑线则为一条从 $(0,0)$ 到 $(4,4)$ 的合法路径。



路径模型 2

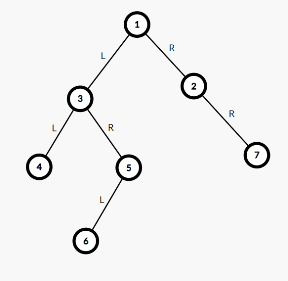
我们可以将每一次从 (x, y) 移动到 $(x, y+1)$ 转化为一个左括号，将从 (x, y) 移动到 $(x+1, y)$ 转化为一个右括号，则每一条从 $(0,0)$ 到 (n, n) 的合法路径都可以唯一对应一个长度为 $2n$ 的括号序列。
上图就转化为括号序列就是 ()(())()。

出入栈序列

现有一个 $p_i = i$ 的排列，每次可以入栈或出栈，出入栈序列则与一个括号序列等价。入栈看作左括号，出栈看作右括号。

二叉树

有 n 个节点的无标号有根的，且区分左右子树的二叉树。每一个节点可以的括号序列为 $(+S_L+)+S_R$。其中 S_L 表示左子树的括号序列，S_R 表示右子树的括号序列，$+$ 表示字符串的连接。如果子树为空，则括号序列为空。



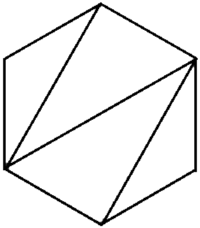
二叉树

上图的节点对应的括号序列入下表所示。

节点编号	括号序列
1	((())(()))()()
2	()()
3	(())(())
4	()
5	(())
6	()
7	()

凸多边形的最优三角形划分

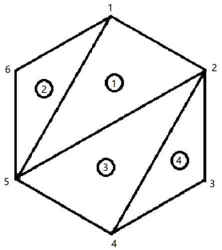
凸多边形的最优三角形划分（以下简称多边形划分）。三角形划分就是指用一些三角形将一个图形不重不漏的覆盖。对于有 n 条边的凸多边形，最少要用 $n-2$ 个三角形划分。最优三角形划分就是指用最少的三角形将图形划分。下图即为一个 6 边形的最优三角形划分。

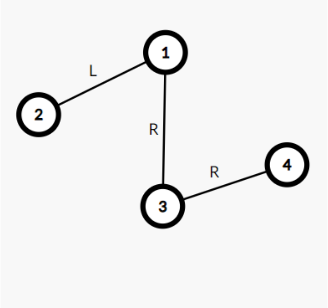


凸多边形的最优三角形划分

我们将每个三角形中建一个节点，再将含有编号为 1 的顶点与编号为 2 的顶点的三角形内的节点作为根，左上方作为左子树，右下方作为右子树，于是就建出了有 $n-2$ 个节点的一棵二叉树。同时，每一棵二叉树也可以用这种方法转化为一种多边形划分。所以棵二叉树与多边形划分是一一对应的。二叉树与括号序列也一一对应，所以每一种多边形划分与括号序列一一对应。

凸多边形的最优三角形划分





上图就是一种标号方式和构建出的二叉树。

计算方式 - 递推式

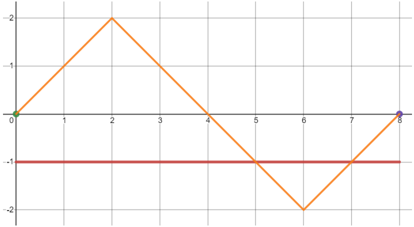
我们设长度为 $2n$ 的括号序列种类数有 F_n 个，那么我们可以得出 F 的前几项 $F_0 = 1, F_1 = 1, F_2 = 2\ldots$。每一个长度为 $2n$ 的括号序列，都可以被写成 $(S_1)(S_2)(S_3)\ldots$ 的形式，其中 S_i 为合法的括号序列。那么它也可以被写成形如 $(A)B$ 的形式，其中 A 和 B 都是一个合法的括号序列，而且每一个括号序列可以刚好对应一组 A, B。
那么我们就可以写出 F 的递推式:

$$F_n = F_0F_{n-1} + F_1F_{n-2} + \cdots + F_{n-1}F_0 = \sum_{i=0}^{n-1} F_iF_{n-i-1}$$

由此，我们就可以在 $O(n^2)$ 的时间内计算出 F_0 至 F_n 的值。

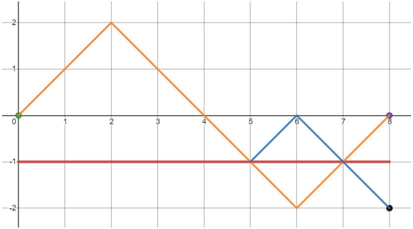
计算方式 - 组合数

上面介绍的路径模型 1 可以用来推导组合数计算方式。由定义得，从 $(0,0)$ 到 $(2n,0)$ 的合法路径条数即为 F_n。易得从 $(0,0)$ 到 $(2n,0)$ 的总路径数为从 $2n$ 个选择中选出 n 个向上（向下）的方案数，即为 C_{2n}^n。我们发现不合法的路径都与 $y=-1$ 这条直线相交。如图即为一条不合法路径。



计算方式 - 组合数

我们想求不合法路径的数量，就可以将不合法路径的集合与一个好求数量的集合建立双射。我们可以将第一次接触到 $y=-1$ 这条线后的部分以 $y=-1$ 对称。



计算方式 - 组合数

如图，蓝色线即为对称后的线。我们就将每一个不合法路径对应到了一条到 $(2n,-2)$ 的路径。显然所有不合法路径和从 $(0,0)$ 到 $(2n,-2)$ 的路径一一对应。因此不合法路径的数量就是从 $(0,0)$ 到 $(2n,-2)$ 的路径数量，也就是从 $2n$ 个选择中选出 $n-1$ 个向上的方案数，就是 C_{2n}^{n-1}。
综上所述，$F_n = C_{2n}^{n} - C_{2n}^{n-1}$。
由此，我们就可以在 $O(n)$ 的时间内计算出 F_0 至 F_n 的值。

例题

洛谷 P8725 画中漂流
线性做法

例题

洛谷 P7263 Something Comforting

数论基础

OI-Wiki

gcd、exgcd

OI-Wiki

裴蜀定理

OI-Wiki

ax+by=c 的整数通解

令 $g=\gcd(a,b)$。接下来我们令 $a\leftarrow\frac{a}{g}$，$b\leftarrow\frac{b}{g}$，$c\leftarrow\frac{c}{g}$，这对解没有影响。

那么，通解为 $x=x_0+kb$，$y=y_0-ka$，其中 (x_0,y_0) 是 $ax+by=c$ 的一个特解，k 是任意整数。

证明则大概是，考虑任意一个解 (x,y)，有 $a(x-x_0)+b(y-y_0)=0$，所以 $a(x-x_0)=-b(y-y_0)$。由于 $\gcd(a,b)=1$，那么若 $x-x_0$ 不是 b 的倍数，$a(x-x_0)$ 也不可能是 b 的倍数。于是 $x-x_0$ 为 b 倍数，设为 kb，计算得知其满足通解形式。

模板：洛谷 P5656

欧拉函数

OI-Wiki

费马小定理

OI-Wiki

模意义下的运算体系

加、减、乘、乘方都是容易定义的。
对于除法，则需要用乘法逆元的方式定义。我们的目的是将乘上逆元等价为乘法的逆操作，也即乘 a 再乘 a 逆元不变。那么就可以将逆元定义为满足方程 $ax \equiv 1 \pmod m$ 的 x。也即，$ax + km = 1$ 的在 $[0, m)$ 范围内的解。可以发现 $\gcd(a, m) = 1$ 时该方程有唯一解。
而对于乘方的两种逆操作，开方和对数，就非常复杂了。模意义下开方过于数学，一般也不会考。而取对数则可以用原根与 BSGS 解决，也几乎不会考。如果愿意学习，可以看 OI-Wiki 的“剩余”界面和“离散对数”界面。
现在是 2025 年了，恐怖数论板子都退环境了。

威尔逊定理

OI-Wiki
(可能其实没什么用)
例题：Genshin OI Round 2 Fermat-1
题解
(题外话：本场比赛的第二题 Fermat-2，很符合我对原神的想像)

中国剩余定理

OI-Wiki
不推荐使用原中国剩余定理。建议使用合并方程的方式。
模板题：洛谷 P4777

Lucas 定理

OI-Wiki
例题：洛谷 P6669/P8688 组合数问题

Thank You

北斗学友