數學

侯欣緯

2023 師大附中暑期資訊培訓

大綱

- 快速冪
- 數論
- 矩陣
- 組合

快速冪

快速冪

CSES Exponentiation

求 $a^b \bmod 10^9+7$。
$0 \le a, b \le 10^9$

快速冪

- 把 a 乘 b 次？
- TLE

快速冪

- 把 a 乘 b 次？
- TLE
- $13 = 2^0 + 2^2 + 2^3$，$a^{13} = a^{2^0} \times a^{2^2} \times a^{2^3}$
- 把 b 二進位分解

快速冪

- 把 a 乘 b 次？
- TLE
- $13 = 2^0 + 2^2 + 2^3$，$a^{13} = a^{2^0} \times a^{2^2} \times a^{2^3}$
- 把 b 二進位分解
- 一開始令 $ans \leftarrow 1$
- 從 b 的最低位開始讀，如果是 1 的話就 $ans \leftarrow ans \times a$
- 到下一位的時候，讓 $a \leftarrow a^2$，這樣可以保持在讀到第 k 位時 a 是一開始的 a^{2^k}
- 時間複雜度 $O(\log b)$

Code

```
const ll MOD = 1000000007;
ll fp(ll a, ll b){
    ll ans = 1;
    while(b > 0){
        if(b & 1) ans = ans * a % MOD;
        a = a * a % MOD;
        b >>= 1;
    }
    return ans;
}
```

數論

判斷質數

- 質數：因數只有 1 和自己的數
- 從 2 檢查到 $n-1$ 看是否整除：$O(n)$

判斷質數

- 質數：因數只有 1 和自己的數
- 從 2 檢查到 $n-1$ 看是否整除：$O(n)$
- 一個合數 x 必定會有一個不超過 $\sqrt{x}$ 的因數
- 只要檢查 2 到 $\sqrt{n}$ 就好了，時間 $O(\sqrt{n})$

找 n 以內的質數

- 暴力：$n\sqrt{n}$

找 n 以內的質數

- 暴力：$n\sqrt{n}$
- 如果我們找到了一個質數，我們就知道它的倍數都不是質數
- 從小的數字開始，如果沒有被標成不是質數的話，它就是質數
- 找到質數就去把它的倍數都標成不是質數

找 n 以內的質數

- 暴力：$n\sqrt{n}$
- 如果我們找到了一個質數，我們就知道它的倍數都不是質數
- 從小的數字開始，如果沒有被標成不是質數的話，它就是質數
- 找到質數就去把它的倍數都標成不是質數
- 時間複雜度 $O(\frac{n}{2}+\frac{n}{3}+\frac{n}{5}+\dots)=O(n\log\log n)$
- 如果刪的時候是從第 i 倍開始的話，可以做一點常數優化
-

```
vector<bool> isprime(n + 1, true);
vector<int> prime;
for(int i = 2; i <= n; i++){
    if(!isprime[i]) continue;
    prime.eb(i);
    for(int j = i; i * j <= n; j++){
        isprime[i * j] = false;
    }
}
```

線性篩

- $O(n \log \log n)$ 的質數篩，每個合數被刪到的次數是不重複的質因數數量
- 能不能每個都只刪到一次？

線性篩

- $O(n \log \log n)$ 的質數篩，每個合數被刪到的次數是不重複的質因數數量
- 能不能每個都只刪到一次？
- 對所有的數（不只質數），去刪它的「不超過最小質因數的質數」倍的數
- 如果目前數字是 x，刪它的 p 倍的時候，p 肯定是 px 的最小質因數
- 假設 x 的最小質因數是 p，那它只會被 $\frac{x}{p}$ 刪到
- 每個人都只被刪到一次，所以時間是 $O(n)$
-

```
vector<int> lpf(n, 1); //每個數的最小質因數
vector<int> prime; //找到的質數

for(int i = 2; i <= n; i++){
    if(lpf[i] == 1){
        lpf[i] = i;
        prime.eb(i);
    }
    for(int j : prime){
        if(i * j > n) break;
        lpf[i * j] = j;
        if(j == lpf[i]) break;
    }
}
```

質因數分解

- 暴力找質因數：$O(n)$
- 其實 x 最多只有一個大於 $\sqrt{x}$ 的質因數
- 先找到 $\sqrt{x}$ 以內的所有質因數，除掉後剩下的就是最大那個質因數
- 時間 $O(\sqrt{n})$

質因數分解

- x 最多只有 $\lfloor \log_2 x \rfloor$ 個質因數
- 如果可以知道每個數的最小質因數，就可以 $O(\log n)$ 解決了

質因數分解

- x 最多只有 $\lfloor \log_2 x \rfloor$ 個質因數
- 如果可以知道每個數的最小質因數，就可以 $O(\log n)$ 解決了
- 先線性篩，得到值域內的數的最小質因數
- 時間 $O(n + q \log n)$，如果不需要多次詢問的話用 $O(\sqrt{n})$ 的作法就好了
-

```
vector<pii> pf; //質因數分解的結果，每一項記為（底數，指數）

while(x > 1){
    int d = lpf[x];
    pf.eb(mp(d, 0));
    while(x % d == 0){
        pf.back().second++;
        x /= d;
    }
}
```

練習

CSES Counting Divisors

有 n 筆詢問，每筆詢問求 x 的因數個數。
$n \leq 10^5$、$x \leq 10^6$

輾轉相除法

引理

對於 $a \geq b$，$\gcd(a, b) = \gcd(a - b, b)$ $(\gcd(a, 0) = a)$。

證明： $d \mid a \wedge d \mid b \iff d \mid (a - b) \wedge d \mid b$，所以 a、b 的公因數和 a、$a - b$ 是一樣的。

輾轉相除法

- 也叫歐幾里得演算法
- $\gcd(a, b) = \gcd(a - b, b) = \gcd(b, a \bmod b)$
- 遞迴直到 b 變成 0 為止
- 用迴圈寫常數比較小
- 其實不用自己寫，有 `__gcd()` 可以用

輾轉相除法

- 也叫歐幾里得演算法
- $\gcd(a, b) = \gcd(a - b, b) = \gcd(b, a \bmod b)$
- 遞迴直到 b 變成 0 為止
- 用迴圈寫常數比較小
- 其實不用自己寫，有 `__gcd()` 可以用
- $a \bmod b \leq \frac{a}{2}$，每過兩次 a 會至少變小一半
- 時間 $O(\log \max(a, b))$

Code

```
int gcd(int a, int b){
    while(b > 0){
        int t = a % b;
        a = b;
        b = t;
    }
    return a;
}
```

擴展歐幾里得演算法

二元一次求整數解

求 $ax + by = c$ 的 x、y 皆為整數的解。

貝祖定理

定理 (貝祖定理)

對於整數 a、b、c，$ax + by = c$ 有整數解若且唯若 $\gcd(a, b) \mid c$。

擴展歐幾里得演算法

- 先找 $ax + by = \gcd(a, b) = g$ 的解
- 我們知道 $\gcd(a, b) = \gcd(b, a \bmod b)$
- 試著把它變成一樣的形狀

$$\begin{aligned} ax + by &= \gcd(a, b) \\ &= \gcd(b, a \bmod b) \\ &= \gcd(b, a - \lfloor \frac{a}{b} \rfloor b) \\ &= bx' + (a - \lfloor \frac{a}{b} \rfloor b)y' \\ &= ay' + b(x' - \lfloor \frac{a}{b} \rfloor y') \end{aligned}$$

- 先求 $bx' + (a - \lfloor \frac{a}{b} \rfloor b)y'$ 的解
- $x = y'$、$y = x' - \lfloor \frac{a}{b} \rfloor y'$
- 遞迴處理，$b = 0$ 的時候回傳 $x = 1$、$y = 0$

Code

```
pii exgcd(int a, int b){
    if(b == 0) return mp(1, 0);
    pii ans = exgcd(b, a % b);
    return mp(ans.S, ans.F - a / b * ans.S);
}
```

負數小細節

- 通常在說的 $a \bmod b$、$\gcd(a, b)$ 都是正的
- 不過常常出現係數是負的狀況
- __gcd、% 算出來的東西有正有負（自己測試）
- 只要有地方的正負號不一致就會出事

負數小細節

- 通常在說的 $a \bmod b$、$\gcd(a,b)$ 都是正的
- 不過常常出現係數是負的狀況
- `__gcd`、`%` 算出來的東西有正有負（自己測試）
- 只要有地方的正負號不一致就會出事
- 前面寫的式子裡，$a \bmod b = a - \lfloor \frac{a}{b} \rfloor b$ 是恆正的
- 但 C++ 裡 `a % b` 實際上是 `a - a / b * b`
- 所以如果你不小心把 `a / b` 寫成下高斯，遞迴的時候還是寫 `exgcd(b, a % b)` 就會出事，除非你寫 `exgcd(b, a - a / b * b)`
- 可是這樣會造成結果的正負號會跟 `__gcd(a, b)` 不一樣

負數小細節

- 通常在說的 $a \bmod b$、$\gcd(a,b)$ 都是正的
- 不過常常出現係數是負的狀況
- `__gcd`、`%` 算出來的東西有正有負（自己測試）
- 只要有地方的正負號不一致就會出事
- 前面寫的式子裡，$a \bmod b = a - \lfloor \frac{a}{b} \rfloor b$ 是恆正的
- 但 C++ 裡 `a % b` 實際上是 `a - a / b * b`
- 所以如果你不小心把 `a / b` 寫成下高斯，遞迴的時候還是寫 `exgcd(b, a % b)` 就會出事，除非你寫 `exgcd(b, a - a / b * b)`
- 可是這樣會造成結果的正負號會跟 `__gcd(a, b)` 不一樣
- 真的不喜歡有負數的話，可以把係數的負號移到變數上
- 例如 $-3x + y = 1$ 可以變成 $3(-x) + y = 1$
- 注意後面用到 $\gcd(a,b)$ 的時候，可能是正的或負的

其他整數解

- $g = \gcd(a, b)$、$a' = \frac{a}{g}$、$b' = \frac{b}{g}$
- 有兩組整數解 (x_1, y_1)、(x_2, y_2)
- $ax_1 + by_1 = ax_2 + by_2 \implies a'(x_1 - x_2) = b'(y_2 - y_1)$
- 因為 a'、b' 互質，所以 $x_1 - x_2 = kb'$、$y_2 - y_1 = ka'$
- 結論是如果找到一組整數解 (x, y)，那其他的整數解都可以表示成 $(x + kb', y - ka')$

貝祖定理告訴我們的更多事

- 除了前面說的以外，貝祖定理還告訴我們
  - 一定存在恰兩組整數解 (x, y) 滿足 $|x| \leq |b'|$、$|y| \leq |a'|$
  - 擴展歐幾里得演算法得到的一定是其中一組
- 所以在你想要最小正數解之類的時候，不用移動很多次
- 還有算的時候不用擔心溢位，計算過程都會在範圍裡

歐拉函數

- $\phi(n) =$ 不超過 n 且與 n 互質的正整數數量

$$\phi(n) = n \prod_{p|n} \frac{p-1}{p} = n \prod_{p|n} \left(1 - \frac{1}{p}\right)$$

- 質因數分之質因數減 1 再乘上 n
- 在數學課學排容的時候可能會遇到它 (?)

模除

- $a \bmod m = r$，r 是 a 除以 m 時的餘數
- 同餘：$a \equiv b \pmod{m}$，表示 $a \bmod m = b \bmod m$，等價於 $m \mid (a - b)$
- $a \equiv a + km \pmod{m}$，$k \in \mathbb{Z}$
- $a \equiv b \pmod{m} \implies a + c \equiv b + c \pmod{m}$
- $a \equiv b \pmod{m} \implies a - c \equiv b - c \pmod{m}$
- $a \equiv b \pmod{m} \implies ac \equiv bc \pmod{m}$
- 「模 m 下」可以任意加、乘（減法是一種加）而不改變同餘關係，且我們希望數字都在 $[0, m)$ 內
- 模下加法和乘法也有交換、結合、分配律

不同除法定義下的模

	$4 \bmod 3$	$(-4) \bmod 3$	$4 \bmod (-3)$	$(-4) \bmod (-3)$
向零取整	1	-1	-1	1
向下取整	1	2	2	1
向上取整	-2	-1	-1	-2

- C++ 裡的 `/` 和 `%` 是向零取整
- 但我們喜歡數字都是正的
- $a \bmod b$ 要寫成 `(a % b + b) % b`

一些定理

- 歐拉定理：$a^{\phi(n)} \equiv 1 \pmod{n}$，$\gcd(a, n) = 1$
- 費馬小定理：$a^p \equiv a \pmod{p}$，p 是質數
- 在 a 不是 p 的倍數的時候，$a^{p-1} \equiv 1 \pmod{p}$、$a^n \equiv a^{n \bmod (p-1)} \pmod{p}$

模逆元

- 在模下可以加、減、乘，但不能像平常的除法那樣除
- 那有沒有類似除法的東西？

模逆元

- 在模下可以加、減、乘，但不能像平常的除法那樣除
- 那有沒有類似除法的東西？
- 除法是乘法的逆運算
- $a \times \frac{1}{a} = 1$
- 我們想找個一樣的東西

模逆元

- 在模下可以加、減、乘，但不能像平常的除法那樣除
- 那有沒有類似除法的東西？
- 除法是乘法的逆運算
- $a \times \frac{1}{a} = 1$
- 我們想找個一樣的東西
- $aa^{-1} \equiv 1 \pmod{m}$
- 也就是 $aa^{-1} - mt = 1$，(a^{-1}, t) 是未知數，根據貝祖定理，a、m 要互質
- 模逆元在 $[0, m)$ 是唯一的

找模逆元

- 費馬小定理：$a^{-1} \equiv a^{p-2} \pmod{p}$，用在模數是質數的狀況（例如 10^9）
- 歐拉定理：$a^{-1} \equiv a^{\phi(n)-1} \pmod{p}$，用在 n 不大的狀況
- 擴展歐幾里得演算法：$aa^{-1} - mt = 1$ 求 a^{-1}

線性建表

- $i^{-1} \equiv (m - \lfloor \frac{m}{i} \rfloor) \times (m \bmod i)^{-1} \pmod m$
- 證明：

$$
\begin{aligned}
m - \lfloor \frac{m}{i} \rfloor \times i &\equiv m \bmod i \pmod m \\
m \times i - \lfloor \frac{m}{i} \rfloor \times i &\equiv m \bmod i \pmod m \\
i \times (m - \lfloor \frac{m}{i} \rfloor) &\equiv m \bmod i \pmod m \\
i \times (m - \lfloor \frac{m}{i} \rfloor) \times (m \bmod i)^{-1} &\equiv 1 \pmod m \\
(m - \lfloor \frac{m}{i} \rfloor) \times (m \bmod i)^{-1} &\equiv i^{-1} \pmod m
\end{aligned}
$$

- 直接遞迴也很快，不知道為什麼

練習

- ZeroJudge a289 Modular Multiplicative Inverse
- CSES Exponentiation II

矩陣

定義

- 矩陣 $=$ 二維陣列
- 列（row）是橫的、行（column）是直的
- 有 n rows & m columns 的矩陣大小記作 $n \times m$
- 在第 i row、第 j column 的元素記作 $M_{i,j}$

$$M = \begin{bmatrix} M_{1,1} & M_{1,2} & M_{1,3} \\ M_{2,1} & M_{2,2} & M_{2,3} \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

基本運算

- A, B, C 是同大小矩陣
- 加法：$A + B = C$，$C_{i,j} = A_{i,j} + B_{i,j}$
- 減法：$A - B = C$，$C_{i,j} = A_{i,j} - B_{i,j}$
- 純量乘法 $n \times A = B$，$B_{i,j} = n \times A_{i,j}$
- 加法有交換、結合律，純量乘法對加法有分配律

矩陣乘法

- A：$n \times m$ 的矩陣
- B：$m \times p$ 的矩陣
- C：$n \times p$ 的矩陣
- $A \times B = C$，$C_{i,j} = \sum_{k=1}^{m} A_{i,k} \times B_{k,j}$
- 有結合律但沒有交換律
- 硬做的時間複雜度是 $O(nmp)$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 4 & 8 \\ 7 & 6 \\ 3 & 0 \end{bmatrix} = \begin{bmatrix} 27 & 20 \\ 69 & 62 \end{bmatrix}$$

矩陣快速冪

- 數字可以快速冪，矩陣也要快速冪！
- 等一下，數字快速冪的時候我們一開始會讓答案 $= 1$，那矩陣裡的 1 是什麼？

矩陣快速冪

- 數字可以快速冪，矩陣也要快速冪！
- 等一下，數字快速冪的時候我們一開始會讓答案 $= 1$，那矩陣裡的 1 是什麼？
- 單位矩陣：I_n 是 $n \times n$ 的矩陣，滿足對於任意同大小的矩陣 A，$AI = IA = A$
- 不難發現到 I_n 主對角線是 1、其他都是 0
- 本來就只有正方形矩陣可以算冪

線性遞迴

- 還記得費氏數列嗎？
- $f(n) = f(n-1) + f(n-2)$，$f(1) = f(2) = 1$
- DP 的話要 $O(n)$ 才能算出 $f(n)$

線性遞迴

- 還記得費氏數列嗎？
- $f(n) = f(n-1) + f(n-2)$，$f(1) = f(2) = 1$
- DP 的話要 $O(n)$ 才能算出 $f(n)$
- 線性遞迴：轉移式是一些係數乘上狀態的一次方的遞迴
- 線性遞迴可以寫成矩陣

$$\begin{bmatrix} f(n) \\ f(n-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} f(n-1) \\ f(n-2) \end{bmatrix}$$

- 把後面展開

$$\begin{bmatrix} f(n) \\ f(n-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^2 \begin{bmatrix} f(n-2) \\ f(n-3) \end{bmatrix} = \cdots = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{n-1} \begin{bmatrix} f(2) \\ f(1) \end{bmatrix}$$

- 快速冪 $O(\log n)$ 解決

例題

CSES Graph Paths I

給一張 n 個節點的有向圖，求從節點 1 走恰 k 步到節點 n 的方法數。答案 $\bmod 10^9+7$ 後輸出。

$n \leq 100$、$k \leq 10^9$

例題

- 令 W 是給的圖的鄰接矩陣，也就是 $W_{i,j} = 1$ 表示存在邊 (i, j)，反之 $W_{i,j} = 0$
- 令 $dp[c][i][j] =$ 從 i 走恰 c 步到 j 的方法數

例題

- 令 W 是給的圖的鄰接矩陣，也就是 $W_{i,j} = 1$ 表示存在邊 (i, j)，反之 $W_{i,j} = 0$
- 令 $dp[c][i][j] =$ 從 i 走恰 c 步到 j 的方法數
- $dp[1][i][j] = W_{i,j}$、$\forall c > 1,\ dp[c][i][j] = \sum_t dp[c-1][i][t] \times W_{t,j}$
- 把 $dp[c]$ 當矩陣看，$dp[c] = dp[c-1] \times W$
- $dp[c] = W^c$

練習

TIOJ 2053 ／ 2018 入營考 pC 費氏數列

給你 x_1, x_2, a, b, n，定義 $\forall i \geq 3,\ x_i = bx_{i-1} + ax_{i-2}$，求 x_n。
$0 \leq x_1, x_2, a, b \leq 10^9$、$3 \leq n \leq 10^9$

練習

2020 北市賽 pC

有 6 種棋子，每一種都有無限個，你要取 n 個排成一列，其中第一種棋子和第二種棋子必須出現偶數次，求有幾種排法。

$n \leq 10^9$

練習

TIOJ 2151 ／ 2020 建中校內賽複試 pB 大富翁

給一張 N 點 M 邊的有向圖，一開始你在節點 s，且開心度是 0，每走一步開心度就會增加 c_i，i 是你走到的節點，c_i 可能是負的，求走了恰 t 步之後開心度最大是多少。

$N \leq 200$、$t \leq 10^{14}$

高斯消去法

- 你有 n 個 n 元一次方程式，怎麼求 $(x_1, x_2, \ldots, x_n)$？

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n = b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n = b_2 \\ \qquad\qquad\qquad\qquad \vdots \\ a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{n,n}x_n = b_n \end{cases}$$

- 加減消去法？
- 高斯消去法 $==$ 加減消去法進化版

高斯消去法

- 基本列運算：
  - 把矩陣的某兩個 row 交換
  - 把矩陣的某個 row 乘上某個數
  - 把矩陣的某個 row 加上另一個 row 的倍數
- 要解方程組，就先把係數都塞進一個矩陣裡

$$\left[\begin{array}{cccc|c} a_{1,1} & a_{1,2} & \dots & a_{1,n} & b_1 \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} & b_2 \\ & & \vdots & & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} & b_n \end{array}\right]$$

直線左側叫係數矩陣，這整個矩陣叫增廣矩陣

- 用基本列運算把直線左側消到主對角線都是 1、其它都是 0
- 最後直線右邊就是答案

詳細步驟

- 假設現在在對一個 $n \times m$ 的矩陣高斯消去
- 從第一 column 處理到第 m column，假設現在在做第 i 個
- 假設第一個還沒固定的 row 是第 r row
- 找到還沒固定的 row 中，第 i column 非 0 的 row，把它和第 r row 交換，找不到就跳過這個 column
- 把第 r row 的第一個非 0 項變成 1
- 拿第 r row 去把其他所有 row 的第 i column 消成 0
- 把第 r row 固定，做下一個 column

結果

- 全 0 的 row 都在最下面
- 每一 row 的第一個非 0 項位置遞增
- 每一 row 的第一個非 0 項是 1，且那個 column 其他位置都是 0

結果

- 全 0 的 row 都在最下面
- 每一 row 的第一個非 0 項位置遞增
- 每一 row 的第一個非 0 項是 1，且那個 column 其他位置都是 0
- 你會得到一個 reduced row echelon form
- 時間複雜度是 $O(n^2m)$

結果

- 全 0 的 row 都在最下面
- 每一 row 的第一個非 0 項位置遞增
- 每一 row 的第一個非 0 項是 1，且那個 column 其他位置都是 0
- 你會得到一個 reduced row echelon form
- 時間複雜度是 $O(n^2m)$
- 如果有一個 row 係數的部分都是 0，等號右邊不是，那就無解
- 沒有無解的話，那有全 0 row 就代表有無限多解
- 確定有解的話，第 i row 最後一項就是答案

關於精度

- 答案有可能會是浮點數
- 不怕溢位的話，在消的時候可以取 LCM
- 通常題目不會太刁難，要嘛就數字小，不然就是帶模

Code

```
void elimination(vector<vector<ll>>& v){
    int n = v.size(), m = v[0].size();
    int r = 0;
    for(int i = 0; i < m; i++){
        bool ok = false;
        for(int j = r; j < n; j++){
            if(v[j][i] == 0) continue;
            v[r].swap(v[j]);
            ok = true;
            break;
        }
        if(!ok) continue;
        ll tmp = inv(v[r][i]);
        for(int j = 0; j < m; j++) v[r][j] = v[r][j] * tmp % MOD;
        for(int j = 0; j < n; j++){
            if(j == r) continue;
            ll now = v[j][i];
            for(int k = 0; k < m; k++){
                v[j][k] -= v[r][k] * now % MOD;
                v[j][k] = (v[j][k] % MOD + MOD) % MOD;
            }
        }
        r++;
    }
}
```

練習

TIOJ 2170 ／ 2019 北市賽 pD 地圖編修

帶模聯立方程組求解，保證唯一解。

關於高斯消去

- 其實上面寫的叫 Gauss-Jordan Elimination
- 高斯消去法只會把下三角消掉
- 不過打比賽的人好像沒有特別分這兩個 (?)
- 有時候不消上三角可以省一點時間
- 高斯消去的其他用途：
  - 算 rank（就是消完後的非全 0 row 數量）
  - 算行列式
  - 算反矩陣
  - 線性基和各種線性代數怪東東

行列式

- 方陣 $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ 的行列式記作 $\det(A) = |A| = \begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix}$
- 令 $|A| = v$，對 A 做某些操作後的 $|A|$：
  - 轉置：v
  - 把某兩 row 或 column 交換：$-v$
  - 把某個 row ／ column 乘上某個數後加到另一個 row ／ column：v
  - 把某個 row ／ column 變成 t 倍：tv

計算

- 一階行列式 $|a| = a$
- A 是 n 階方陣
- 令 $B_{i,j}$ 是 A 去掉第 i row、第 j column 的 $n-1$ 階方陣
- 展開：
  - 選某個 i，$|A| = \sum_{j=1}^{n} (-1)^{i+j} A_{i,j} |B_{i,j}|$
  - 選某個 j，$|A| = \sum_{i=1}^{n} (-1)^{i+j} A_{i,j} |B_{i,j}|$
- 例如

$$\begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} = 1 \times 4 - 2 \times 3$$

計算

- 蛤可是剛剛那個的時間複雜度是 $T(n) = nT(n-1) + O(n) = O(n!)$ 欸
- 對 A 高斯消去，記錄操作對 $|A|$ 改變，算消完的結果的行列式再推回去
- 對第一個 column 展開，因為只有 $A_{1,1}$ 不是 0 所以只要算 $A_{1,1}|B_{1,1}|$ 就好
- 其實就是把對角線乘起來
- 高斯消去後的對角線要嘛是 0 要嘛是 1
- 也可以不把首項變 1

反矩陣

- A 是 n 階方陣
- $AA^{-1} = A^{-1}A = I$，A^{-1} 是 A 的反矩陣
- 把矩陣乘法展開的話，你會拿到 n^2 個式子
- 其實有一些的係數是一樣的，所以可以把它們塞在一起變成 $[A \mid I]$
- 例如

$$\left[\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 3 & 4 & 0 & 1 \end{array}\right]$$

- 對左半部高斯消去到變成 I，右半部就是答案
- 如果有全 0 row，也就是 $|A| = 0$ 的話，A^{-1} 不存在

組合

基本公式

- $n!$ 表示有 n 個相異物品排成一列的方法數。
- n^m 表示有 m 個相異物品和 n 種狀態，每個物品必須有恰一種狀態的方法數。
- $C_k^n = \frac{n!}{k!(n-k)!}$ 表示 n 個相異物品中選 k 個的方法數。
- $P_k^n = \frac{n!}{(n-k)!}$ 表示 n 個相異物品選 k 個並排成一列的方法數。
- $H_k^n = C_k^{n+k-1} = \frac{(n+k-1)!}{k!(n-1)!}$ 表示 n 個相異物品選 k 個，同一個物品可以重複選的方法數。
- 至於怎麼用就是個人造化了 (X

巴斯卡三角形

$$
\begin{array}{ccccccccc}
 & & & 1 & & & & & \\
 & & 1 & & 1 & & & & \\
 & 1 & & 2 & & 1 & & & \\
1 & & 3 & & 3 & & 1 & & \\
1 & & 4 & & 6 & & 4 & & 1
\end{array}
=
\begin{array}{ccccc}
 & & C_0^0 & & \\
 & C_0^1 & & C_1^1 & \\
C_0^2 & & C_1^2 & & C_2^2 \\
C_0^3 & & C_1^3 & & C_2^3 \quad C_3^3 \\
C_0^4 & C_1^4 & & C_2^4 & C_3^4 \quad C_4^4
\end{array}
$$

巴斯卡定理

- $C_m^n = C_m^{n-1} + C_{m-1}^{n-1}$
- 展開後會變前綴和
- 會出現在一些 $O(n^2)$ DP 優化成 $O(n)$ 的地方

例題

選位子

有 n 個座位排成一列，有 m 個人，每個人都要坐在一個坐位上。一個坐位只能坐一個人，且兩個人不能坐在相鄰的座位，求有幾種坐法。

$1 \leq m \leq n \leq 10^6$

先想個 DP

- $dp[i][j] =$ 左到右第 i 個人坐在第 j 個位置的方法數
- $dp[i][j] = \sum_{k=1}^{j-2} dp[i-1][k]$
- 大概長下面那樣
- 其實它是多階前綴和往後移 $2(i-1)$ 格
- $1, 1, 1, \ldots$ 的多階前綴和就是巴斯卡三角形

	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2	0	0	1	2	3	4	5	6
3	0	0	0	0	1	3	6	10
4	0	0	0	0	0	0	1	4

排容原理

$$\left|\bigcup_{1\le i\le n} S_i\right| = \sum_{1\le i\le n} |S_i| - \sum_{1\le i<j\le n} |S_i \cap S_j| + \sum_{1\le i<j<k\le n} |S_i \cap S_j \cap S_k|$$
$$- \cdots + (-1)^{n-1} |S_1 \cap S_2 \cap \cdots \cap S_n|$$

練習

AtCoder ABC172E NEQ

求 $A_1, A_2, \ldots, A_N$ 、$B_1, B_2, \ldots, B_N$ 的對數，滿足

- $\forall 1 \leq i \leq N,\ 1 \leq A_i, B_i \leq M \wedge A_i \neq B_i$
- $\forall 1 \leq i < j \leq N,\ A_i \neq A_j \wedge B_i \neq B_j$

答案 $\bmod 10^9 + 7$ 後輸出。

$N \leq M \leq 5 \times 10^5$

卡特蘭數

對角線走法問題

有 $n \times n$ 的棋盤，你只能在格線上走，左下角的格子點是 $(0,0)$ 右上角是 (n,n)，你一開始在 $(0,0)$，只能往右或往上走，求有幾種方法，在不到 $y = x$ 左上半部的情況下（可以碰到），走到 (n,n)。

卡特蘭數遞迴版

- $C_n =$ 答案
- 試著用遞迴算 C_{n+1}
- 假設除了 $(n+1, n+1)$ 最後一次碰到 $y = x$ 是在 (i, i)
- $(0, 0)$ 到 (i, i) 的方法數顯然就是 C_i

卡特蘭數遞迴版

- $C_n =$ 答案
- 試著用遞迴算 C_{n+1}
- 假設除了 $(n+1, n+1)$ 最後一次碰到 $y = x$ 是在 (i, i)
- $(0, 0)$ 到 (i, i) 的方法數顯然就是 C_i
- 因為是最後一次，所以在 (i, i) 到 $(n+1, n+1)$ 的路徑上沒有碰到 $y = x$
- 那段路會長這樣：$(i, i) \to (i+1, i) \to (n+1, n) \to (n+1, n+1)$
- $(i+1, i) \to (n+1, n)$ 不會越過 $y = x - 1$，可以看成是 C_{n-i}
- $C_{n+1} = \sum_{i=0}^{n} C_i C_{n-i}$

卡特蘭數公式版

- 換另一種角度，這次用刪去法算 C_n
- 不合法的路徑一定會碰到 $y = x + 1$
- 把 $(0, 0)$ 到第一次碰到 $y = x + 1$ 的部分對稱到 $y = x + 1$ 的另一邊
- 變成一條從 $(-1, 1)$ 到 (n, n) 的路徑

卡特蘭數公式版

- 換另一種角度，這次用刪去法算 C_n
- 不合法的路徑一定會碰到 $y = x + 1$
- 把 $(0, 0)$ 到第一次碰到 $y = x + 1$ 的部分對稱到 $y = x + 1$ 的另一邊
- 變成一條從 $(-1, 1)$ 到 (n, n) 的路徑
- 不合法路徑和 $(-1, 1)$ 到 (n, n) 的路徑一一對應
- $(0, 0) \to (n, n)$ 有 C_n^{2n} 種、$(-1, 1) \to (n, n)$ 有 C_{n-1}^{2n} 種
- $C_n = C_n^{2n} - C_{n-1}^{2n} = \frac{(2n)!}{n!(n+1)!} = \frac{1}{n+1} C_n^{2n}$

卡特蘭數

$$C_0 = 1,\ C_n = \sum_{i=0}^{n-1} C_i C_{n-i-1} = C_n^{2n} - C_{n-1}^{2n}$$

- 這就是卡特蘭數
- 很多看起來沒什麼關係的東西，都跟它有關
- n 對括號組成的合法括號字串有 C_n 種
- n 個節點的二元樹（左右有別）有 C_n 種
- 還有很多很多

裸題練習

CSES Bracket Sequences I

求長度是 n 的合法括號字串數量 $\bmod 10^9+7$。
$n \leq 10^6$