数论&组合数学常见结论及例题

球盒模型（八种）

参考链接。给定 n 个小球 m 个盒子。

- 球同，盒不同、不能空

> 隔板法： N 个小球即一共 $N-1$ 个空，分成 M 堆即 $M-1$ 个隔板，答案为 $\binom{n-1}{m-1}$。

- 球同，盒不同、能空

> 隔板法：多出 $M-1$ 个虚空球，答案为 $\binom{m-1+n}{n}$。

- 球同，盒同、能空

> $\frac{1}{(1-x)(1-x^2)\dots(1-x^m)}$ 的 x^n 项的系数。动态规划，答案为
>
> $$dp[i][j]=\begin{cases}dp[i][j-1]+dp[i-j][j] & i\ge j\\ dp[i][j-1] & i<j\\ 1 & j==1\,||\,i\le 1\end{cases}$$

```
int dp[15][15];
void choose(int n) {
    for(int i = 0; i < n; i ++) {
        for(int j = 1; j < n; j ++) {
            if(i <= 1 || j == 1) dp[i][j] = 1;
            else if (i < j) dp[i][j] = dp[i][j - 1];
            else dp[i][j] = dp[i][j - 1] + dp[i - j][j];
        }
    }
}
```

- 球同，盒同、不能空

> $\frac{x^m}{(1-x)(1-x^2)\dots(1-x^m)}$ 的 x^n 项的系数。动态规划，答案为
>
> $$dp[n][m]=\begin{cases}dp[n-m][m] & n\ge m\\ 0 & n<m\end{cases}$$

- 球不同，盒同、不能空

> 第二类斯特林数 $\texttt{Stirling2}(n,m)$ ，答案为
>
> $$dp[n][m]=\begin{cases}m*dp[n-1][m]+dp[n-1][m-1] & 1\le m<n\\ 1 & 0\le n==m\\ 0 & m==0且1\le n\end{cases}$$

- 球不同，盒同、能空

> 第二类斯特林数之和 $\sum_{i=1}^{m}\texttt{Stirling2}(n,m)$ ，答案为 $\sum_{i=0}^{m}dp[n][i]$。

- 球不同，盒不同、不能空

> 第二类斯特林数乘上 m 的阶乘 $m! \cdot \texttt{Stirling2}(n,m)$ ，答案为 $dp[n][m]*m!$ 。

- 球不同，盒不同、能空

> 答案为 m^n 。

麦乐鸡定理

给定两个互质的数 n,m ，定义 $x=a*n+b*m$ （$a\ge 0,b\ge 0$），当 $x>n*m-n-m$ 时，该式子恒成立。

抽屉原理（鸽巢原理）

将 $n+1$ 个物体，划分为 n 组，那么有至少一组有两个（或以上）的物体。

哥德巴赫猜想

任何一个大于 5 的整数都可写成三个质数之和；任何一个大于 2 的偶数都可写成两个素数之和。

除法、取模运算的本质

有公式：$x\div i=\left\lfloor\frac{x}{i}\right\rfloor+x-i\cdot\left\lfloor\frac{x}{i}\right\rfloor$ ，$x \mod i=x-i\cdot\left\lfloor\frac{x}{i}\right\rfloor$ 。

与、或、异或

运算	运算符、数学符号表示	解释
与	`&`、`and`	同1出1
或	`|`、`or`	有1出1
异或	`^`、$\bigoplus$、`xor`	不同出1

一些结论：

> 对于给定的 X 和序列 $[a_1,a_2,\dots,a_n]$ ，有：$\boldsymbol{X=(X\&a_1)or(X\&a_2)or\dots or(X\&a_n)}$ 。
> 原理是 and 意味着取交集，or 意味着取子集。[来源 - 牛客小白月赛49C]

调和级数近似公式

```
log(n) + 0.5772156649 + 1.0 / (2 * n)
```

欧拉函数常见性质

- $1-n$ 中与 n 互质的数之和为 $n*\varphi(n)/2$ 。
- 若 $a,\ b$ 互质，则 $\varphi(a*b)=\varphi(a)*\varphi(b)$ 。实际上，所有满足这一条件的函数统称为积性函数。
- 若 f 是积性函数，且有 $n=\prod\limits_{i=1}^{m}p_i^{c_i}$ ，那么 $f(n)=\prod\limits_{i=1}^{m}f(p_i^{c_i})$ 。
- 若 p 为质数，且满足 $p\mid n$ ，
  - $p^2\mid n$ ，那么 $\varphi(n)=\varphi(n/p)*p$ 。

  - $p^2 \nmid n$，那么 $\varphi(n) = \varphi(n/p) * (p-1)$ 。
- $\sum\limits_{d|n} \varphi(d) = n$ 。

> 如 $n = 10$ ，则 $d = 10/5/2/1$ ，那么 $10 = \varphi(10) + \varphi(5) + \varphi(2) + \varphi(1)$ 。

- $\sum\limits_{i=1}^{n} \gcd(i, n) = \sum\limits_{d|n} \left\lfloor \frac{n}{d} \right\rfloor \varphi(d)$ （欧拉反演）。

组合数学常见性质

- $k * C_n^k = n * C_{n-1}^{k-1}$ ；
- $C_k^n * C_m^k = C_m^n * C_{m-n}^{m-k}$ ；
- $C_n^k + C_n^{k+1} = C_{n+1}^{k+1}$ ；
- $\sum_{i=0}^{n} C_n^i = 2^n$ ；
- $\sum_{k=0}^{n} (-1)^k * C_n^k = 0$ 。
- 二项式反演：$\begin{cases} f_n = \sum\limits_{i=0}^{n} \binom{n}{i} g_i \Leftrightarrow g_n = \sum\limits_{i=0}^{n} (-1)^{n-i} \binom{n}{i} f_i \\ f_k = \sum\limits_{i=k}^{n} \binom{i}{k} g_i \Leftrightarrow g_k = \sum\limits_{i=k}^{n} (-1)^{i-k} \binom{i}{k} f_i \end{cases}$ ；
- $\sum\limits_{i=1}^{n} i \binom{n}{i} = n * 2^{n-1}$ ；
- $\sum\limits_{i=1}^{n} i^2 \binom{n}{i} = n * (n+1) * 2^{n-2}$ ；
- $\sum\limits_{i=1}^{n} \frac{1}{i} \binom{n}{i} = \sum\limits_{i=1}^{n} \frac{1}{i}$ ；
- $\sum\limits_{i=0}^{n} \binom{n}{i}^2 = \binom{2n}{n}$ ；
- 拉格朗日恒等式：$\sum\limits_{i=1}^{n} \sum\limits_{j=i+1}^{n} (a_i b_j - a_j b_i)^2 = (\sum\limits_{i=1}^{n} a_i)^2 (\sum\limits_{i=1}^{n} b_i)^2 - (\sum\limits_{i=1}^{n} a_i b_i)^2$ 。

lucas定理

Lucas 定理内容如下：对于质数 p，有

$$\binom{n}{m} \bmod p = \binom{\lfloor n/p \rfloor}{\lfloor m/p \rfloor} \cdot \binom{n \bmod p}{m \bmod p} \bmod p$$

观察上述表达式，可知 $n \bmod p$ 和 $m \bmod p$ 一定是小于 p 的数，可以直接求解，$\binom{\lfloor n/p \rfloor}{\lfloor m/p \rfloor}$ 可以继续用 Lucas 定理求解。这也就要求 p 的范围不能够太大，一般在 10^5 左右。边界条件：当 $m = 0$ 的时候，返回 1。

时间复杂度为 $O(f(p) + g(n) \log n)$，其中 $f(n)$ 为预处理组合数的复杂度，$g(n)$ 为单次求组合数的复杂度。

```cpp
long long Lucas(long long n, long lm, long long p) {
  if (m == 0) return 1;
  return (C(n % p, m % p, p) * Lucas(n / p, m / p, p)) % p;
}
```

```python
def Lucas(n, m, p):
    if m == 0:
        return 1
    return (C(n % p, m % p, p) * Lucas(n // p, m // p, p)) % p
```

范德蒙德卷积公式

在数量为 $n+m$ 的堆中选 k 个元素，和分别在数量为 n、m 的堆中选 i、$k-i$ 个元素的方案数是相同的，即 $\sum_{i=0}^{k}\binom{n}{i}\binom{m}{k-i}=\binom{n+m}{k}$；

变体：

- $\sum_{i=0}^{k} C_{i+n}^{i}=C_{k+n+1}^{k}$；
- $\sum_{i=0}^{k} C_{n}^{i}*C_{m}^{i}=\sum_{i=0}^{k} C_{n}^{i}*C_{m}^{m-i}=C_{n+m}^{n}$。

推论 1 及证明

$$\sum_{i=-r}^{s}\binom{n}{r+i}\binom{m}{s-i}=\binom{n+m}{r+s}$$

推论 2 及证明

$$\sum_{i=1}^{n}\binom{n}{i}\binom{n}{i-1}=\sum_{i=0}^{n-1}\binom{n}{i+1}\binom{n}{i}=\sum_{i=0}^{n-1}\binom{n}{n-1-i}\binom{n}{i}=\binom{2n}{n-1}$$

推论 3 及证明

$$\sum_{i=0}^{n}\binom{n}{i}^{2}=\sum_{i=0}^{n}\binom{n}{i}\binom{n}{n-i}=\binom{2n}{n}$$

推论 4 及证明

$$\sum_{i=0}^{m}\binom{n}{i}\binom{m}{i}=\sum_{i=0}^{m}\binom{n}{i}\binom{m}{m-i}=\binom{n+m}{m}$$

其中 $\binom{n+m}{m}$ 是我们较为熟悉的网格图路径计数的方案数。所以我们可以考虑其组合意义的证明。

在一张网格图中，从 $(0,0)$ 走到 (n,m) 共走 $n+m$ 步。规定 $(0,0)$ 位于网格图左上角，其中向下走了 n 步，向右走了 m 步，方案数为 $\binom{n+m}{m}$。

换个视角，我们将 $n+m$ 步拆成两部分走，先走 n 步，再走 m 步，那么 n 步中若有 i 步向右，则 m 步中就有 $m-i$ 步向右，故得证。

卡特兰数

是一类奇特的组合数，前几项为 $1,1,2,5,14,42,132,429,1430,4862$ 。如遇到以下问题，则直接套用即可。

- 【括号匹配问题】 n 个左括号和 n 个右括号组成的合法括号序列的数量，为 Cat_n 。
- 【进出栈问题】 $1,2,\dots,n$ 经过一个栈，形成的合法出栈序列的数量，为 Cat_n 。
- 【二叉树生成问题】 n 个节点构成的不同二叉树的数量，为 Cat_n 。
- 【路径数量问题】在平面直角坐标系上，每一步只能**向上**或**向右**走，从 $(0,0)$ 走到 (n,n) ，并且除两个端点外不接触直线 $y=x$ 的路线数量，为 $2Cat_{n-1}$ 。

计算公式：$Cat_n=\dfrac{C_{2n}^n}{n+1}$ ，$Cat_n=\dfrac{Cat_{n-1}*(4n-2)}{n+1}$ 。

狄利克雷卷积

$\sum\limits_{d|n}\varphi(d)=n$ ，$\sum\limits_{d|n}\mu(d)\dfrac{n}{d}=\varphi(n)$ 。

斐波那契数列

通项公式：$F_n=\dfrac{1}{\sqrt5}*\left[\left(\dfrac{1+\sqrt5}{2}\right)^n-\left(\dfrac{1-\sqrt5}{2}\right)^n\right]$ 。

直接结论：

- 卡西尼性质：$F_{n-1}*F_{n+1}-F_n^2=(-1)^n$ ；
- $F_n^2+F_{n+1}^2=F_{2n+1}$ ；
- $F_{n+1}^2-F_{n-1}^2=F_{2n}$ （由上一条写两遍相减得到）；
- 若存在序列 $a_0=1,a_n=a_{n-1}+a_{n-3}+a_{n-5}+\dots(n\ge1)$ 则 $a_n=F_n(n\ge1)$ ；
- 齐肯多夫定理：任何正整数都可以表示成若干个不连续的斐波那契数（ F_2 开始）可以用贪心实现。

求和公式结论：

- 奇数项求和：$F_1+F_3+F_5+\dots+F_{2n-1}=F_{2n}$ ；
- 偶数项求和：$F_2+F_4+F_6+\dots+F_{2n}=F_{2n+1}-1$ ；
- 平方和：$F_1^2+F_2^2+F_3^2+\dots+F_n^2=F_n*F_{n+1}$ ；
- $F_1+2F_2+3F_3+\dots+nF_n=nF_{n+2}-F_{n+3}+2$ ；
- $-F_1+F_2-F_3+\dots+(-1)^nF_n=(-1)^n(F_{n+1}-F_n)+1$ ；
- $F_{2n-2m-2}(F_{2n}+F_{2n+2})=F_{2m+2}+F_{4n-2m}$ 。

数论结论：

- $F_a\mid F_b\Leftrightarrow a\mid b$ ；
- $\gcd(F_a,F_b)=F_{\gcd(a,b)}$ ；
- 当 p 为 $5k\pm1$ 型素数时，$\begin{cases}F_{p-1}\equiv0\pmod p\\F_p\equiv1\pmod p\\F_{p+1}\equiv1\pmod p\end{cases}$ ；
- 当 p 为 $5k\pm2$ 型素数时，$\begin{cases}F_{p-1}\equiv1\pmod p\\F_p\equiv-1\pmod p\\F_{p+1}\equiv0\pmod p\end{cases}$ ；

- $F(n)\%m$ 的周期 $\le 6m$ （$m=2\times 5^k$ 时取到等号）；
- 既是斐波那契数又是平方数的有且仅有 $1,144$ 。

杂

- 负数取模得到的是负数，如果要用 $0/1$ 判断的话请取绝对值；
- 辗转相除法原式为 $\gcd(x,y)=\gcd(x,y-x)$ ，推广到 N 项为 $\gcd(a_1,a_2,\dots,a_N)=\gcd(a_1,a_2-a_1,\dots,a_N-a_{N-1})$ ，
  - 该推论在"四则运算后 gcd "这类题中有特殊意义，如求解 $\gcd(a_1+X,a_2+X,\dots,a_N+X)$ 时See；
- 以下式子成立：$\gcd(a,m)=\gcd(a+x,m)\Leftrightarrow \gcd(a,m)=\gcd(x,m)$ 。求解上式满足条件的 x 的数量即为求比 $\dfrac{m}{\gcd(a,m)}$ 小且与其互质的数的个数，即用欧拉函数求解 $\varphi\left(\dfrac{m}{\gcd(a,m)}\right)$ 。
- 已知序列 a ，定义集合 $S=\{a_i\cdot a_j\mid i<j\}$ ，现在要求解 $\gcd(S)$ ，即为求解 $\gcd(a_j,\gcd(a_i\mid i<j))$ ，换句话说，即为求解后缀 gcd 。
- 连续四个数互质的情况如下，当 n 为奇数时，$n,n-1,n-2$ 一定互质；而当 n 为偶数时，$\begin{cases} n,n-1,n-3\text{互质} & \gcd(n,n-3)=1\text{时}\\ n-1,n-2,n-3\text{互质} & \gcd(n,n-3)\neq 1\text{时}\end{cases}$ See；
- 由 $a\mod b=(b+a)\mod b=(2\cdot b+a)\mod b=\dots=(K\cdot b+a)\mod b$ 可以推广得到 $(a\mod b)\mod c=((K\cdot bc+a)\mod b)\mod c$ ，由此可以得到一个 bc 的答案周期See；
- 对于长度为 $2\cdot N$ 的数列 a ，将其任意均分为两个长度为 N 的数列 p,q ，随后对 p 非递减排序、对 q 非递增排序，定义 $f(p,q)=\sum\limits_{i=1}^{n}|p_i-q_i|$ ，那么答案为 a 数列前 N 大的数之和减去前 N 小的数之和See。
- 令 $\begin{cases}X=a+b\\ Y=a\oplus b\end{cases}$ ，**如果**该式子**有解**，那么存在前提条件 $\begin{cases}X\ge Y\\ X,Y\text{同奇偶}\end{cases}$ ；进一步，此时最小的 a 的取值为 $\dfrac{X-Y}{2}$ See。

  然而，上方方程并不总是有解的，只有当变量增加到三个时，才**一定有解**，即：**在保证上方前提条件成立的情况下**，求解 $\begin{cases}X=a+b+c\\ Y=a\oplus b\oplus c\end{cases}$ ，则一定存在一组解 $\{\dfrac{X-Y}{2},\dfrac{X-Y}{2},Y\}$ See。
- 已知序列 p 是由序列 a_1 、序列 a_2 、……、序列 a_n 合并而成，且合并过程中各序列内元素相对顺序不变，记 $T(p)$ 是 p 序列的最大前缀和，则 $T(p)=\sum\limits_{i=1}^{n}T(a_i)$ See 。
- $x+y=x|y+x\&y$ ，对于两个数字 x 和 y ，如果将 x 变为 $x|y$ ，同时将 y 变为 $x\&y$ ，那么在本质上即将 x 二进制模式下的全部 1 移动到了 y 的对应的位置上 See 。
- 一个正整数 x 异或、加上另一个正整数 y 后奇偶性不发生变化：$a+b\equiv a\oplus b(\mod 2)$ See 。

常见例题

题意：将 1 至 N 的每个数字分组，使得每一组的数字之和均为质数。输出每一个数字所在的组别，且要求分出的组数最少 See 。

考察哥德巴赫猜想，记全部数字之和为 S ，分类讨论如下：

- 为 S 质数时，只需要分入同一组；
- 当 S 为偶数时，由猜想可知一定能分成两个质数，可以证明其中较小的那个一定小于 N ，暴力枚举分组；
- 当 $S-2$ 为质数时，特殊判断出答案；

- 其余情况一定能被分成三组，其中 3 单独成组，$S-3$ 后成为偶数，重复讨论二的过程即可。

题意：给定一个长度为 n 的数组，定义这个数组是 BAD 的，当且仅当可以把数组分成两个子序列，这两个子序列的元素之和相等。现在你需要删除**最少的**元素，使得删除后的数组不是 BAD 的。

最少删除一个元素——如果原数组存在奇数，则直接删除这个奇数即可；反之，我们发现，对数列同除以一个数不影响计算，故我们只需要找到最大的满足 $2^k \mid a_i$ 成立的 2^k，随后将全部的 a_i 变为 $\dfrac{a_i}{2^k}$，此时一定有一个奇数（换句话说，我们可以对原数列的每一个元素不断的除以 2 直到出现奇数为止），删除这个奇数即可 See 。

题意：设当前有一个数字为 x，减去、加上最少的数字使得其能被 k 整除。

最少减去 $x \bmod k$ 这个很好想；最少加上 $\left(\left\lceil\dfrac{x}{k}\right\rceil * k\right) \bmod k$ 也比较好想，但是更简便的方法为加上 $k - x \bmod k$，这个式子等价于前面这一坨。

题意：给定一个整数 n，用恰好 k 个 2 的幂次数之和表示它。例如：$n=9, k=4$，答案为 $1+2+2+4$。

结论1：k 合法当且仅当 `__builtin_popcountll(n) <= k && k <= n`，显然。

结论2：$2^{k+1} = 2 \cdot 2^k$，所以我们可以将二进制位看作是数组，然后从高位向低位推，一个高位等于两个低位，直到数组之和恰好等于 k，随后依次输出即可。举例说明，$\{1,0,0,1\} \to \{0,2,0,1\} \to \{0,1,2,1\}$，即答案为 0 个 2^3、1 个 2^2、……。

```
signed main() {
    int n, k;
    cin >> n >> k;

    int cnt = __builtin_popcountll(n);

    if (k < cnt || n < k) {
        cout << "NO\n";
        return 0;
    }
    cout << "YES\n";

    vector<int> num;
    while (n) {
        num.push_back(n % 2);
        n /= 2;
    }

    for (int i = num.size() - 1; i > 0; i--) {
        int p = min(k - cnt, num[i]);
        num[i] -= p;
        num[i - 1] += 2 * p;
        cnt += p;
    }

    for (int i = 0; i < num.size(); i++) {
        for (int j = 1; j <= num[i]; j++) {
            cout << (1LL << i) << " ";
```

```
        }
    }
}
```

题意：n 个取值在 $[0, k)$ 之间的数之和为 m 的方案数

答案为 $\sum_{i=0}^{n} -1^i \cdot \binom{n}{i} \cdot \binom{m-i \cdot k+n-1}{n-1}$ See1 See2。

```
Z clac(int n, int k, int m) {
    Z ans = 0;
        ans += C(n, i) * C(m - i * k + n - 1, n - 1) * pow(-1, i);
    }
    return ans;
}
```

[1] 先考虑没有 k 的限制，那么即球盒模型：m 个球放入 n 个盒子，球同、盒子不同、能空。使用隔板法得到公式：`C(m + n - 1, n - 1)`；[2] 下面加上取值范围后进一步考虑：假设现在 n 个数之和为 $m-k$，运用上述隔板法可得公式：`C(m - k + n - 1, n - 1)`；[3] 随后，选择任意一个数字，将其加上 k，这样，这个数字一定不满足条件，选法为：`C(n, 1)`；[4] 此时，至少有一个数字是不满足条件的，按照一般流程，到这里，`C(m + n - 1, n - 1) - C(n, 1) * C(m - k + n - 1, n - 1)` 即是答案；但是，这样的操作会导致重复的部分，所以这里要使用容斥原理将重复部分去除（关于为什么会重复，试比较概率论中的加法公式）。

/END/