第一章

热身题

1 所有的马都有同样的颜色，我们可以对给定集合中的马匹数量运用归纳法来证明之．理由就是："如果恰有一匹马，那么它与它自身有相同的颜色，故而基础是显然的．根据归纳法的步骤，假设有 n 匹马，标号从1到 n．根据归纳假设，标号从1直到 $n-1$ 的马都有同样的颜色，类似地，标号从2直到 n 的马也有同样的颜色．但是，处于中间位置标号从2直到 $n-1$ 的马，当它们在不同的马群中时不可能改变颜色，因为这些是马，而不是变色龙．故而依据传递性可知，标号从1直到 n 的马也必定有同样的颜色，于是全部 n 匹马都有同样的颜色．证毕．"如果这一推理有误，那么错在哪儿？

错在基本情况,当n>2的时候,没问题;当n=2时,会出现问题.

2 把有 n 个圆盘的塔从左边的桩柱A移动到右边的桩柱B，不允许在A和B之间直接移动，求最短的移动序列．（每一次移动都必须是移动到中间的桩柱或者从中间的桩柱移出．像通常一样，较大的圆盘永远不能放在较小圆盘的上面．）

考虑移动n个圆盘从A到B的过程：

1. 首先将n-1个较小圆盘从A移动到B（这需要T_{n-1}次移动）。
2. 然后将最大的圆盘从A移动到M（1次移动）。
3. 接着将n-1个圆盘从B移动回A（这需要T_{n-1}次移动），以便B为空。
4. 将最大的圆盘从M移动到B（1次移动）。
5. 最后将n-1个圆盘从A移动到B（这需要T_{n-1}次移动）。

总移动次数为： T_n = T_{n-1} + 1 + T_{n-1} + 1 + T_{n-1} = 3T_{n-1} + 2

求解即可,答案是

求解可得

$$
\begin{aligned}
T_n + 1 &= 3(T_{n-1} + 1) \\
&= 3^{n-1}(T_1 + 1) \\
&= 3^n \\
T_n &= 3^n - 1
\end{aligned}
$$

3 证明，在上一题限制下的移动过程中，实际上，我们会在3根桩柱上都遇到 n 个圆盘的每一种正确的叠放．

备注：

正确的叠放是指，将n个圆盘分为三组，那么每组的顺序从小到大的。

记

- $T(n)$: 在2的限制下,我们会在3根桩柱上都遇到n个圆盘的每一种正确的叠放。

关于n做数学归纳法即可。

当$k=1$时，k的移动路径为$A\to C\to B$，所以结论成立。

假设当$k=n-1$时结论成立，下面证明$k=n$时结论也成立。

考虑之前的移动步骤：

步骤	A	B	C
0	$[1,\ldots,n]$	空	空
1	$[n]$	$[1,\ldots,n-1]$	空
2	空	$[1,\ldots,n-1]$	$[n]$
3	$[1,\ldots,n-1]$	空	$[n]$
4	$[1,\ldots,n-1]$	$[n]$	空
5	空	$[1,\ldots,n]$	空

考虑n在的位置：

如果n在A，由归纳假设可知将$[1,\ldots,n-1]$从A移动到B会遇到每一种正确的叠放，这也对应了n在A情形下每一种正确的叠放。n在B,C的情形同理可得。所以$k=n$时结论也成立。

4　是否存在 n 个圆盘在3根桩柱上的某种开始叠放或结束叠放，使得按照卢卡斯原来的规则，需要多于 2^n-1 次的移动？

记

- $T(n)$: 对于n个圆盘的任意摆列方式，按照卢卡斯原来的规则，最多需要2^n-1次移动将圆盘移动到B。

关于n做数学归纳法即可。

当$n=1$时结论显然。

假设当$n=m-1$时结论成立，下面证明$n=m$时结论也成立。

如果m在B，那么对剩余$n-1$个圆盘运用$T(n-1)$即可，移动次数

$$\le 2^{m-1}-1<2^m-1$$

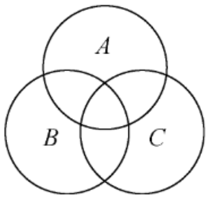
如果m不在B，由对称性不妨设m在A，那么移动过程如下：

- $[1,\ldots,m-1]\to C$
- $[m]\to B$
- $[1,\ldots,m-1]\to B$

所以移动次数

$$\le 1+2^{m-1}-1+2^{m-1}-1\le 2^m-1$$

5 由3个重叠的圆做成的维恩图常用来描述与3个给定集合有关的8个可能的子集：



由4个给定集合给出的16种可能的子集能否用4个重叠的圆来描述？

两个圆最多相交于两点，所以第4个圆最多和前3个圆交于6点，最多增加6块区域，所以总区域最多为

$$6+8=14$$

补充说明：

答案中给的卵形和圆的最大不同之处在于，两个卵形可以有4个交点。

6 平面上由 n 条直线定义的某些区域是无界的，而另一些区域则是有界的. 有界区域的最大个数是多少？

记n条直线产生的有界区域最大个数为a_n。

考虑第n条直线，该直线最多和$n-1$条直线相交，即产生$n-1$个交点，相邻两个点最多可以增加一个有界区域，所以最多增加$n-2$个有界区域，因此

$$\begin{aligned}
a_n &= a_{n-1}+n-2\\
a_1 &= a_2 = 0\\
a_n &= a_1+\sum_{i=0}^{n-2} i\\
&= \frac{(n-2)(n-1)}{2}
\end{aligned}$$

7 设 $H(n)=J(n+1)-J(n)$. 方程（1.8）告诉我们有 $H(2n)=2$，而对 $n\geqslant 1$ 有

$$H(2n+1)=J(2n+2)-J(2n+1)=\bigl(2J(n+1)-1\bigr)-\bigl(2J(n)+1\bigr)=2H(n)-2.$$

于是，看起来有可能通过对 n 用归纳法，证明对所有 n 都有 $H(n)=2$. 这里什么地方有错？

$$\begin{aligned}
H(1) &= J(2)-J(1)\\
&= 2J(1)-1-J(1)\\
&= 0
\end{aligned}$$

即base case结论都不成立，所以无法使用数学归纳法。

课后题

8　解递归式

$$Q_0=\alpha\,;\qquad Q_1=\beta\,;$$
$$Q_n=(1+Q_{n-1})/Q_{n-2}\,,\qquad n>1\,.$$

假设对所有 $n\geqslant 0$ 都有 $Q_n\neq 0$．提示：$Q_4=(1+\alpha)/\beta$．

$$\begin{aligned}
Q_0 &= \alpha \\
Q_1 &= \beta \\
Q_2 &= \frac{1+Q_1}{Q_0} \\
&= \frac{1+\beta}{\alpha} \\
Q_3 &= \frac{1+Q_2}{Q_1} \\
&= \frac{1+\frac{1+\beta}{\alpha}}{\beta} \\
&= \frac{1+\alpha+\beta}{\alpha\beta} \\
Q_4 &= \frac{1+Q_3}{Q_2} \\
&= \frac{1+\frac{1+\alpha+\beta}{\alpha\beta}}{\frac{1+\beta}{\alpha}} \\
&= \frac{1+\alpha+\beta+\alpha\beta}{(1+\beta)\beta} \\
&= \frac{1+\alpha}{\beta} \\
Q_5 &= \frac{1+\frac{1+\alpha}{\beta}}{\frac{1+\alpha+\beta}{\alpha\beta}} \\
&= \alpha \\
Q_6 &= \frac{1+Q_5}{Q_4} \\
&= \frac{1+\alpha}{\frac{1+\alpha}{\beta}} \\
&= \beta
\end{aligned}$$

所以

$$Q_k = Q_{k-5}, k \geq 5$$

$Q = 0, \dots, 4$的定义如上。所以结论成立

9 有时可以利用反向归纳法，它是从 n 到 $n-1$ 来证明命题，而不是相反！例如，考虑命题

$$P(n): \quad x_1 \cdots x_n \leqslant \left(\frac{x_1 + \cdots + x_n}{n}\right)^n, \quad x_1, \cdots, x_n \geqslant 0.$$

这对 $n=2$ 为真，因为 $(x_1+x_2)^2 - 4x_1x_2 = (x_1-x_2)^2 \geqslant 0$.

a 令 $x_n = (x_1 + \cdots + x_{n-1})/(n-1)$ ，证明只要 $n > 1$ ， $P(n)$ 就蕴涵 $P(n-1)$.

b 证明 $P(n)$ 和 $P(2)$ 蕴涵 $P(2n)$.

c 说明为什么这就蕴涵了 $P(n)$ 对所有 n 为真.

对于题目a:

假设$P(n)$成立，令

$$x_n = (x_1 + \dots + x_{n-1})/(n-1)$$

那么

$$\begin{aligned}
\frac{x_1 + \dots + x_n}{n} &= \frac{x_1 + \dots + x_{n-1} + (x_1 + \dots + x_{n-1})/(n-1)}{n} \\
&= \frac{x_1 + \dots + x_{n-1}}{n-1}
\end{aligned}$$

所以

$$\begin{aligned}
x_1 \dots x_n \leq (\frac{x_1 + \dots + x_n}{n})^n \Leftrightarrow \\
x_1 \dots x_{n-1} \frac{x_1 + \dots + x_{n-1}}{n-1} \leq (\frac{x_1 + \dots + x_{n-1}}{n-1})^n \Leftrightarrow \\
x_1 \dots x_{n-1} \leq (\frac{x_1 + \dots + x_{n-1}}{n-1})^{n-1}
\end{aligned}$$

所以$P(n) \Rightarrow P(n-1)$。

对于题目b:

如果$P(n), P(2)$成立，那么

$$
\begin{aligned}
\prod_{i=1}^{2n} x_i &= (\prod_{i=1}^{n} x_i) \times (\prod_{i=n+1}^{2n} x_i) \\
&\le (\frac{1}{n}\sum_{i=1}^{n} x_i)^n \times (\frac{1}{n}\sum_{i=n+1}^{2n} x_i)^n \\
&\le ((\frac{1}{n}\sum_{i=1}^{n} x_i) \times (\frac{1}{n}\sum_{i=n+1}^{2n} x_i))^n \\
&\le \left((\frac{1}{2n}\sum_{i=1}^{2n} x_i)^2\right)^n \\
&= (\frac{1}{2n}\sum_{i=1}^{2n} x_i)^{2n}
\end{aligned}
$$

所以$P(2n)$成立。

对于题目c:

首先由$P(2)$成立以及(b)的结论可得$P(2^k)$成立。

接着，对于任意n ⊟ 2^k，总存在k，满足$n < 2^k$，所以由(a)可得

$$P(2^k) \Rightarrow P(2^k - 1) \Rightarrow \ldots \Rightarrow P(n)$$

从而$P(n)$成立。

10 设 Q_n 是将一个有 n 个圆盘的塔从A移动到B所需要的最少移动次数，要求所有的移动都必须是顺时针方向的，也就是说，从A到B，从B到其他的桩柱，再从其他的桩柱到A. 又设 R_n 是在这一限制下从B返回到A所需要移动的最少次数. 证明

$$Q_n = \begin{cases} 0, & n = 0; \\ 2R_{n-1} + 1, & n > 0; \end{cases} \quad R_n = \begin{cases} 0, & n = 0; \\ Q_n + Q_{n-1} + 1, & n > 0. \end{cases}$$

（无需解这些递归式，我们将在第7章里介绍怎样做. ）

$A \to B$:

步骤	A	B	C
0	$[1, \ldots, n]$	空	空
1	$[n]$	空	$[1, \ldots, n-1]$
2	空	$[n]$	$[1, \ldots, n-1]$
3	空	$[1, \ldots, n]$	空

注意到$A \to C, C \to B$和$B \to A$等价，所以

$$Q_n \le 2R_{n-1}+1$$

另一方面，要使得第n个圆盘移动到B，该步骤至少要1次，并且此时前$n-1$个圆盘应该在C，$A\to C$的最少步骤数为R_{n-1}，$C\to B$的最少步骤数为R_{n-1}，所以

$$Q_n \ge 2R_{n-1}+1$$

综上

$$Q_n = \{ \begin{array}{cc} 0, & n=0 \\ 2R_{n-1}+1, & n>0 \end{array}$$

$B\to A$:

步骤	A	B	C
0	空	$[1,\ldots,n]$	空
1	$[1,\ldots,n-1]$	$[n]$	空
2	$[1,\ldots,n-1]$	空	$[n]$
3	空	$[1,\ldots,n-1]$	$[n]$
4	$[n]$	$[1,\ldots,n-1]$	空
5	$[1,\ldots,n]$	空	空

所以

$$\begin{aligned} R_n &\le R_{n-1}+1+Q_{n-1}+1+R_{n-1} \\ &= 2R_{n-1}+1+Q_{n-1}+1 \\ &= Q_n+Q_{n-1}+1 \end{aligned}$$

另一方面，第n个圆盘从B到A必然要经历两步：

- 第一步：$B\to C$，此时前$n-1$个圆盘必然在A，即前$n-1$个圆盘要从B移动到A，这部分的步骤数$\ge R_{n-1}+1$
- 第二步：$C\to A$，此时前$n-1$个圆盘必然在B，即前$n-1$个圆盘要从A移动到B，所以这部分的步骤数$\ge Q_{n-1}+1$
- 第三步：将B上的圆盘移动到A，这步分的步骤数$\ge R_{n-1}$

所以

$$\begin{aligned} R_n &\ge R_{n-1}+1+Q_{n-1}+1+R_{n-1} \\ &= 2R_{n-1}+1+Q_{n-1}+1 \\ &= Q_n+Q_{n-1}+1 \end{aligned}$$

综上

$$R_n = \{ \begin{array}{cc} 0, & n=0 \\ Q_n + Q_{n-1} + 1, & n>0 \end{array}$$

11 双重河内塔包含 $2n$ 个圆盘，它们有 n 个不同的尺寸，每一种尺寸的圆盘有两个．如通常那样，要求每次只能移动一个圆盘，且不能把较大的圆盘放在较小的圆盘上面.

a 如果相同尺寸的圆盘是相互不可区分的，要把一个双重塔从一根桩柱移动到另一根桩柱需要移动多少次?

b 如果在最后的排列中要把所有同样尺寸的圆盘恢复成原来的从上到下的次序，需要移动多少次?
提 示：这是一个难题，实在应该是个"附加题".

(a)

先将前$2n-2$个圆盘移动到C，然后将最后两个圆盘移动到B，最后将C上的圆盘移动到B，所以递推关系为

$$\begin{aligned} A_n &= 2A_{n-1} + 2 \\ A_1 &= 2 \end{aligned}$$

求解可得

$$\begin{aligned} A_n + 2 &= 2(A_{n-1} + 2) \\ &= 2^{n-1}(A_1 + 2) \\ &= 2^{n+1} \\ A_n &= 2^{n+1} - 2 \end{aligned}$$

注意这种移动方式保持尺寸大小$\le n-1$的圆盘顺序不变，尺寸大小为n的圆盘顺序相反（利用归纳法可证)。

(b)

显然

$$B_1 = 3$$

$B \to A$有如下移动方式:

步骤	A	B	C	**步骤数**
0	$[1,-1,\ldots,n,-n]$	空	空	0
1	$[n,-n]$	$[1,-1\ldots,-(n-1),(n-1)]$	空	A_{n-1}
2	$[-n]$	$[1,-1\ldots,-(n-1),(n-1)]$	$[n]$	1

步骤	A	B	C	步骤数
3	$[-n]$	空	$[1,-1\ldots,(n-1),-(n-1),n]$	A_{n-1}
4	空	$[-n]$	$[1,-1\ldots,(n-1),-(n-1),n]$	1
5	$[1,-1\ldots,-(n-1),(n-1),n]$	$[-n]$	$[n]$	A_{n-1}
6	$[1,-1\ldots,-(n-1),(n-1),n]$	$[n,-n]$	空	1
7	空	$[1,-1,\ldots,(n-1),-(n-1),n,-n]$	空	A_{n-1}

所以

$$B_n \le 4A_{n-1}+3$$

另一方面，$[n,-n]$按照顺序从A到B至少需要3次，具体如下：

- $n: A\to C$，要使得这步能移动，前$2n-2$个圆盘必然在B，这部分的步骤数$\ge A_{n-1}+1$
- $-n: A\to B$，要使得这步能移动，前$2n-2$个圆盘必然在C，这部分的步骤数$\ge A_{n-1}+1$
- $n: C\to B$，要使得这步能移动，前$2n-2$个圆盘必然在A，这部分的步骤数$\ge A_{n-1}+1$
- 最后将A上的圆盘移动到B，这部分的步骤数$\ge A_{n-1}$

所以

$$B_n \ge 4A_{n-1}+3$$

有注意按照(a)的方式操作4次，最终顺序必然是原始的顺序，从而上述不等号可以取到等号。

因此

$$B_n = 4A_{n-1}+3 = 2^{n+3}-5$$

选做题

```
● Minimal moves under Lucas rule (Frame-Stewart)
  (N,M)=(11,4) -> 57
  (N,M)=(11,5) -> 35
  (N,M)=(17,4) -> 177
  (N,M)=(17,5) -> 79
  (N,M)=(19,4) -> 241
  (N,M)=(19,5) -> 99
```

约瑟夫问题的变体

现在我们要选择第m个（m不一定小于n）人作为每次的"幸运者"，而不是第2个。

(1) 闭式解推导

对于$m = 3, 4, 5$的情况，我们推导$J(n)$的闭式解：

m = 3的情况

对于$m = 3$的约瑟夫问题，我们有递推关系：

$$J_3(n) = (J_3(n-1) + 3) \bmod n$$

闭式解：

$$J_3(n) = 3n - 2^{\lfloor \log_2(3n) \rfloor} + 1$$

证明： 设$n = 2^k + t$，其中$0 \le t < 2^k$。

当$t = 0$时，$n = 2^k$：

- 经过2^{k-1}轮后，剩余2^{k-1}个人
- 继续这个过程，最终剩余第1个人
- 所以$J_3(2^k) = 1$

当$t > 0$时：

- 经过t轮后，剩余2^k个人
- 此时从第$3t + 1$个人开始
- 所以$J_3(n) = 3t + 1$

m = 4的情况

闭式解：

$$J_4(n) = 4n - 3^{\lfloor \log_3(4n) \rfloor} + 1$$

m = 5的情况

闭式解：

$$J_5(n) = 5n - 4^{\lfloor \log_4(5n) \rfloor} + 1$$

(2) C++程序实现

对于$m = 7, 11, 17, 37$，$n = 20201001$的情况，我们使用高效的算法：

```
#include <iostream>
#include <vector>
#include <chrono>

// 方法1：优化的递推算法
int josephus_optimized(int n, int m) {
    int result = 0;
    for (int i = 2; i <= n; i++) {
        result = (result + m) % i;
    }
    return result + 1; // 转换为1-indexed
}

// 方法2：分块优化算法（适用于大m）
int josephus_chunked(int n, int m) {
    if (m == 1) return n;

    int result = 0;
    int i = 1;

    while (i <= n) {
        int chunk_size = std::min(n - i + 1, (m - result - 1) / (m - 1) + 1);
        if (chunk_size <= 0) chunk_size = 1;

        result = (result + m * chunk_size) % (i + chunk_size - 1);
        i += chunk_size;
    }

    return result + 1;
}

// 方法3：数学公式法（适用于特定m值）
int josephus_formula(int n, int m) {
    if (m == 2) {
        // 经典约瑟夫问题公式
        int l = n - (1 << (31 - __builtin_clz(n)));
        return 2 * l + 1;
    }

    // 对于其他m值，使用递推
    return josephus_optimized(n, m);
}

int main() {
```

```cpp
    int n = 20201001;
    std::vector<int> m_values = {7, 11, 17, 37};

    std::cout << "约瑟夫问题求解 (n = " << n << ")" << std::endl;
    std::cout << "======================================" << std::endl;

    for (int m : m_values) {
        auto start = std::chrono::high_resolution_clock::now();

        int result = josephus_optimized(n, m);

        auto end = std::chrono::high_resolution_clock::now();
        auto duration = std::chrono::duration_cast<std::chrono::milliseconds>(end - start);

        std::cout << "m = " << m << ": J(" << n << ") = " << result
                  << " (耗时: " << duration.count() << "ms)" << std::endl;
    }

    return 0;
}
```

多桩柱汉诺塔（Lucas 规则）选做题

1761636598005

题意

有 M 根桩

$$(p_1, \ldots, p_M)$$

，初始所有 N 个圆盘在 p_1，目标将其在 Lucas 规则下移动到 p_M。问最少步数。

题目要求：

- (1) 提出一般性的等式/不等式；
- (2) 计算 $(N,M)=(11,4),(11,5),(17,4),(17,5),(19,4),(19,5)$ 的精确最少步数，并附源码。

(1) 一般等式/不等式与性质

- **三柱情形（基线）**：$F(N,3)=2^N-1$。
- **单调性**：$F(N,M+1)\le F(N,M)$；$F(N+1,M) > F(N,M)$。
- **上界（Frame–Stewart 递推）/常用算法**：
- $$F(N, M) \le \min_{1\le k<N} (2, F(k, M) + F(N - k, M - 1)), \quad M \ge 4.$$
- **下界**：$F(N,M)\ge 2N-1$。当 $M\ge N$ 时可达成上界 $2N-1$：先将前 $N-1$ 个圆盘各自停在不同的中间柱上，移最大圆盘，再复位。

- **边界条件**：(F(0,M)=0)；(F(1,M)=1)；(F(N,2)=\infty)（无解）；(F(N,3)=2^N-1)。

综上，一个可用于精确求解的广泛接受的关系是（Frame–Stewart）：

$$F(N,M) = \begin{cases} 2^N - 1, & M = 3, \\ \min_{1 \le k < N} \left(2\,F(k,M) + F(N-k,M-1)\right), & M \ge 4. \end{cases}$$

当

$$(M \ge N)$$

时，取等于 (2N-1)。

(2) C++ 源码与指定 (N,M) 的精确数值

下述程序使用记忆化 DP 实现 Frame–Stewart 递推，能在毫秒级算出题目给定的规模。

```
#include <bits/stdc++.h>
using namespace std;

using u64 = unsigned long long;

// 为安全起见，使用 128 位中间量避免乘 2 的潜在溢出（本题规模用 u64 足够）。
static const u64 INF64 = std::numeric_limits<u64>::max()/4;

u64 F(int N, int M, vector<vector<u64>>& dp) {
    if (N == 0) return 0;
    if (N == 1) return 1;
    if (M == 3) return (N >= 64 ? INF64 : ((1ULL << N) - 1ULL)); // N<=63 安全；题目内 N<=19
    if (M >= N) return 2ULL * N - 1ULL; // M 足够多时可达 2N-1

    u64& memo = dp[N][M];
    if (memo != INF64) return memo;

    u64 best = INF64;
    // Frame–Stewart：枚举拆分 k
    for (int k = 1; k < N; ++k) {
        u64 left = F(k, M, dp);
        u64 right = F(N - k, M - 1, dp);
        if (left == INF64 || right == INF64) continue;
        __uint128_t cand = (__uint128_t)left * 2 + right; // 128 位中间量
        if (cand < best) best = (u64)cand;
    }
    memo = best;
    return memo;
}

int main() {
    vector<pair<int,int>> queries = {
        {11,4},{11,5},{17,4},{17,5},{19,4},{19,5}
    };

    int maxN = 0, maxM = 0;
    for (auto [N,M] : queries) { maxN = max(maxN, N); maxM = max(maxM, M); }
```

```
    vector<vector<u64>> dp(maxN + 1, vector<u64>(maxM + 1, INF64));

    cout << "Minimal moves under Lucas rule (Frame-Stewart)" << "\n";
    for (auto [N,M] : queries) {
        u64 ans = F(N, M, dp);
        cout << "(N,M) = (" << N << "," << M << ") -> " << ans << "\n";
    }
    return 0;
}
```

解答:

1761636310180

第2章 和式

2 化简表达式 $x\times([x>0]-[x<0])$.

2

$$x\times([x>0]-[x<0])=\begin{cases}x & x>0\\ 0 & x=0\\ -x & x<0\end{cases}$$

因此

$$x\times([x>0]-[x<0])=|x|$$

11 分部求和的一般法则（2.56）等价于

$$\sum_{0\le k<n}(a_{k+1}-a_k)b_k=a_nb_n-a_0b_0-\sum_{0\le k<n}a_{k+1}(b_{k+1}-b_k),\quad n\geqslant 0.$$

利用分配律、结合律和交换律直接证明这个公式.

11

$$\begin{aligned}\sum_{0\le k<n}(a_{k+1}-a_k)b_k&=\sum_{0\le k<n}a_{k+1}b_k-\sum_{0\le k<n}a_kb_k\\&=\sum_{0\le k<n}a_{k+1}b_k-\sum_{0\le k<n}a_{k+1}b_{k+1}-a_0b_0+a_nb_n\\&=a_nb_n-a_0b_0-\sum_{0\le k<n}a_{k+1}(b_{k+1}-b_k)\end{aligned}$$

19 利用求和因子来求解递归式

$$T_0 = 5;$$
$$2T_n = nT_{n-1} + 3 \times n!, \quad n > 0.$$

19

$$T_0 = 5$$
$$2T_n = nT_{n-1} + 3 \times n!$$

回顾38页：

$$\begin{aligned} a_n &= 2 \\ b_n &= n \\ s_n &= \frac{a_{n-1}a_{n-2}\cdots a_1}{b_n b_{n-1}\cdots b_2} \\ &= \frac{2^{n-1}}{n!} \end{aligned}$$

可得

$$\begin{aligned} \frac{2^n T_n}{n!} &= \frac{2^{n-1}T_{n-1}}{(n-1)!} + 3 \times 2^{n-1} \\ \frac{2^n T_n}{n!} &= \frac{2^0 T_0}{0!} + \sum_{k=0}^{n-1} 3 \times 2^{k-1} \\ \frac{2^n T_n}{n!} &= 5 + 3(2^n - 1) \\ T_n &= \frac{(3 \times 2^n + 2)n!}{2^n} \\ &= 3n! + \frac{n!}{2^{n-1}} \end{aligned}$$

20 试用扰动法计算 $\sum_{k=0}^{n} kH_k$，不过改为推导出 $\sum_{k=0}^{n} H_k$ 的值.

20

定义

$$S_n = \sum_{k=0}^{n} kH_k$$

那么

$$
\begin{aligned}
S_n+(n+1)H_{n+1} &= \sum_{k=1}^{n+1} kH_k \\
&= \sum_{k=0}^{n} (k+1)H_{k+1} \\
&= \sum_{k=0}^{n} kH_{k+1} + \sum_{k=0}^{n} H_{k+1} \\
&= \sum_{k=0}^{n} k\left(H_k+\frac{1}{k+1}\right) + \sum_{k=1}^{n+1} H_k \\
&= \sum_{k=0}^{n} kH_k + \sum_{k=0}^{n} \frac{k}{k+1} + \sum_{k=0}^{n+1} H_k \\
&= S_n + \sum_{k=0}^{n} \left(1-\frac{1}{k+1}\right) + \sum_{k=0}^{n+1} H_k \\
&= S_n + n + 1 - H_{n+1} + \sum_{k=0}^{n+1} H_k \\
&= S_n + n + 1 + \sum_{k=0}^{n} H_k
\end{aligned}
$$

$$\sum_{k=0}^{n} H_k = (n+1)H_{n+1}-(n+1)$$

$$\sum_{k=0}^{n-1} H_k = nH_n - n$$

21 假设 $n \geqslant 0$，用扰动法计算和式 $S_n=\sum_{k=0}^{n}(-1)^{n-k}$， $T_n=\sum_{k=0}^{n}(-1)^{n-k}k$ 以及 $U_n=\sum_{k=0}^{n}(-1)^{n-k}k^2$.

21

和式一：

$$
\begin{aligned}
S_{n+1} &= \sum_{k=0}^{n+1} (-1)^{n+1-k} \\
&= (-1)^{n+1} + \sum_{k=1}^{n+1} (-1)^{n+1-k} \\
&= (-1)^{n+1} + \sum_{k=1}^{n+1} (-1)^{n+1-k} \\
&= (-1)^{n+1} + \sum_{k=0}^{n} (-1)^{n-k} \\
&= (-1)^{n+1} + S_n \\
-S_n + 1 &= \sum_{k=0}^{n} (-1)^{n+1-k} + (-1)^{n+1-n-1} \\
&= \sum_{k=0}^{n+1} (-1)^{n+1-k} \\
&= S_{n+1} \\
-S_n + 1 &= (-1)^{n+1} + S_n \\
S_n &= \frac{1-(-1)^{n+1}}{2} \\
&= \frac{1+(-1)^{n}}{2}
\end{aligned}
$$

和式二:

$$
\begin{aligned}
T_{n+1} &= \sum_{k=0}^{n+1} (-1)^{n+1-k} k \\
&= \sum_{k=1}^{n+1} (-1)^{n+1-k} k \\
&= \sum_{k=0}^{n} (-1)^{n-k} (k+1) \\
&= \sum_{k=0}^{n} (-1)^{n-k} k + \sum_{k=0}^{n} (-1)^{n-k} \\
&= T_n + S_n \\
-T_n + n + 1 &= \sum_{k=0}^{n} (-1)^{n+1-k} k + (-1)^{n+1-n-1} (n+1) \\
&= \sum_{k=0}^{n+1} (-1)^{n+1-k} k \\
&= T_{n+1} \\
-T_n + n + 1 &= T_n + S_n \\
T_n &= \frac{n+1-S_n}{2}
\end{aligned}
$$

和式三:

$$
\begin{aligned}
U_{n+1} &= \sum_{k=0}^{n+1} (-1)^{n+1-k} k^2 \\
&= \sum_{k=1}^{n+1} (-1)^{n+1-k} k^2 \\
&= \sum_{k=0}^{n} (-1)^{n-k} (k+1)^2 \\
&= \sum_{k=0}^{n} (-1)^{n-k} k^2 + 2\sum_{k=0}^{n} (-1)^{n-k} k + \sum_{k=0}^{n} (-1)^{n-k} \\
&= U_n + 2T_n + S_n \\
-U_n + (n+1)^2 &= \sum_{k=0}^{n} (-1)^{n+1-k} k^2 + (-1)^{n+1-n-1}(n+1)^2 \\
&= \sum_{k=0}^{n+1} (-1)^{n+1-k} (k+1)^2 \\
&= U_{n+1} \\
-U_n + (n+1)^2 &= U_n + 2T_n + S_n \\
U_n &= \frac{(n+1)^2 - 2T_n - S_n}{2} \\
&= \frac{(n+1)^2 - (n+1) + S_n - S_n}{2} \\
&= \frac{(n+1)n}{2}
\end{aligned}
$$

22 （不用归纳法）证明拉格朗日恒等式：

$$\sum_{1\leqslant j<k\leqslant n}(a_j b_k - a_k b_j)^2 = \left(\sum_{k=1}^{n} a_k^2\right)\left(\sum_{k=1}^{n} b_k^2\right) - \left(\sum_{k=1}^{n} a_k b_k\right)^2 .$$

事实上，可证明一个关于更一般的二重和式

$$\sum_{1\leqslant j<k\leqslant n}(a_j b_k - a_k b_j)(A_j B_k - A_k B_j)$$

的恒等式.

22

记：

$$S_1 = \sum_{1 \le j < k \le n} (a_j b_k - a_k b_j)(A_j B_k - A_k B_j)$$

$$S_2 = \sum_{1 \le k < j \le n} (a_k b_j - a_j b_k)(A_k B_j - A_j B_k)$$

$$= \sum_{1 \le j < k \le n} (a_j b_k - a_k b_j)(A_j B_k - A_k B_j)$$

将j, k互换，我们得到

$$S_1 = S_2$$

另一方面

$$\begin{aligned}
2S_1 &= S_1 + S_2 \\
&= \sum_{1 \le j, k \le n} (a_j b_k - a_k b_j)(A_j B_k - A_k B_j) \\
&= \sum_{1 \le j, k \le n} (a_j A_j b_k B_k - b_j A_j a_k B_k - a_j B_j b_k A_k + b_j B_j a_k A_k) \\
&= \sum_{k=1}^{n} \sum_{j=1}^{n} a_j A_j b_k B_k - \sum_{k=1}^{n} \sum_{j=1}^{n} b_j A_j a_k B_k - \sum_{k=1}^{n} \sum_{j=1}^{n} a_j B_j b_k A_k + \sum_{k=1}^{n} \sum_{j=1}^{n} b_j B_j a_k A_k \\
&= (\sum_{k=1}^{n} b_k B_k)(\sum_{j=1}^{n} a_j A_j) - (\sum_{k=1}^{n} a_k B_k)(\sum_{j=1}^{n} b_j A_j) - (\sum_{k=1}^{n} b_k A_k)(\sum_{j=1}^{n} a_j B_j) + (\sum_{k=1}^{n} a_k A_k)(\sum_{j=1}^{n} b_j B_j) \\
&= 2(\sum_{k=1}^{n} a_k A_k)(\sum_{k=1}^{n} b_k B_k) - 2(\sum_{k=1}^{n} a_k B_k)(\sum_{k=1}^{n} b_k A_k) \\
S_1 &= (\sum_{k=1}^{n} a_k A_k)(\sum_{k=1}^{n} b_k B_k) - (\sum_{k=1}^{n} a_k B_k)(\sum_{k=1}^{n} b_k A_k)
\end{aligned}$$

因此

$$\sum_{1 \le j < k \le n} (a_j b_k - a_k b_j)(A_j B_k - A_k B_j) = (\sum_{k=1}^{n} a_k A_k)(\sum_{k=1}^{n} b_k B_k) - (\sum_{k=1}^{n} a_k B_k)(\sum_{k=1}^{n} b_k A_k)$$

特别的，取

$$A_j = a_j$$
$$B_j = b_j$$

可得

$$\sum_{1 \le j < k \le n} (a_j b_k - a_k b_j)^2 = (\sum_{k=1}^{n} a_k^2)(\sum_{k=1}^{n} b_k^2) - (\sum_{k=1}^{n} a_k b_k)^2$$

28

28 下面的推导在何处步入歧途？

$$1=\sum_{k\geqslant 1}\frac{1}{k(k+1)}=\sum_{k\geqslant 1}\left(\frac{k}{k+1}-\frac{k-1}{k}\right)$$
$$=\sum_{k\geqslant 1}\sum_{j\geqslant 1}\left(\frac{k}{j}[j=k+1]-\frac{j}{k}[j=k-1]\right)$$
$$=\sum_{j\geqslant 1}\sum_{k\geqslant 1}\left(\frac{k}{j}[j=k+1]-\frac{j}{k}[j=k-1]\right)$$
$$=\sum_{j\geqslant 1}\sum_{k\geqslant 1}\left(\frac{k}{j}[k=j-1]-\frac{j}{k}[k=j+1]\right)$$
$$=\sum_{j\geqslant 1}\left(\frac{j-1}{j}-\frac{j}{j+1}\right)=\sum_{j\geqslant 1}\frac{-1}{j(j+1)}=-1.$$

第二个等号位置已经有错误了，因为分解的两个级数不绝对收敛。

29

29 计算和式 $\sum_{k=1}^{n}(-1)^k k/(4k^2-1)$.

$$\sum_{k=1}^{n}\frac{(-1)^k k}{4k^2-1}=\sum_{k=1}^{n}\frac{(-1)^k k}{(2k-1)(2k+1)}$$
$$=\frac{1}{4}\sum_{k=1}^{n}(-1)^k\left(\frac{1}{2k-1}+\frac{1}{2k+1}\right)$$
$$=\frac{1}{4}\sum_{k=1}^{n}\left(\frac{(-1)^k}{2k-1}-\frac{(-1)^{k+1}}{2k+1}\right)$$
$$=\frac{1}{4}\left(-1-\frac{(-1)^{n+1}}{2n+1}\right)$$
$$=-\frac{1}{4}+\frac{(-1)^n}{8n+4}$$

追加题：前 n 个正整数五次幂和的闭式，至少三种不同方法

Please find the closed form for the sum of the first n positive integers to the power 5 in at least 3 different ways. You might refer to Chap. 2.5 of CM for its 8 methods for the sum of n squares, but method 0, 1 and 6 are not allowed and new methods not mentioned in CM are extremely encouraged.

记

$$S_5(n)=\sum_{k=1}^{n}k^5.$$

结论（统一答案）：

$$\boxed{S_5(n)=\frac{n^2(n+1)^2\left(2n^2+2n-1\right)}{12}}$$

下面给出三种彼此独立的方法（不使用 CM 2.5 的 0、1、6 号方法），且思路尽量不同：

方法A：Stirling 数 + 二项系数求和（组合恒等式法）

利用恒等式（Stirling 数第二类）：

$$k^m=\sum_{r=0}^{m}S(m,r)\,r!\binom{k}{r},\quad S(m,r)\text{为第二类Stirling数}.$$

两侧对 (k=1,\ldots,n) 求和，并用

$$\sum_{k=1}^{n}\binom{k}{r}=\binom{n+1}{r+1}$$

得到

$$S_5(n)=\sum_{r=1}^{5}S(5,r)\,r!\binom{n+1}{r+1}.$$

取

$$S(5,1)=1,S(5,2)=15,S(5,3)=25,S(5,4)=10,S(5,5)=1$$

，故

$$S_5(n)=\binom{n+1}{2}+30\binom{n+1}{3}+150\binom{n+1}{4}+240\binom{n+1}{5}+120\binom{n+1}{6}.$$

将右侧化简即可得到上面的封闭形式 ($\frac{n^2(n+1)^2(2n^2+2n-1)}{12}$)。

方法B：六次差分/多项式拟合（有限差分法）

已知

$$S_5(n)$$

为关于 (n) 的 6 次多项式，且首项系数为

$$dfrac16$$

（因为

$$sum_{k=1}^{n} k^5 \sim \frac{n^6}{6}$$

设

$$S_5(n) = \frac{1}{6}n^6 + an^5 + bn^4 + cn^3 + dn^2 + en.$$

代入 (n=0,1,2,3,4,5) 的精确值（由直接求和或递推得到）解线性方程组；或用有限差分表，令第六差为常数，从而唯一确定系数 (a,b,c,d,e)。解得

$$S_5(n) = \frac{n^2(n+1)^2(2n^2+2n-1)}{12}.$$

方法C：二项式望远镜（用 $(k+1)^6 - k^6$)线性消元）

展开

$$(k+1)^6 - k^6 = 6k^5 + 15k^4 + 20k^3 + 15k^2 + 6k + 1.$$

对 (k=0) 到 (n) 求和，左侧望远镜得$(n+1)^6 - 0^6$。右侧得到

$$(n+1)^6 = 6S_5(n) + 15S_4(n) + 20S_3(n) + 15S_2(n) + 6S_1(n) + (n+1).$$

代入已知

$$S_1(n) = \frac{n(n+1)}{2}, \quad S_2(n) = \frac{n(n+1)(2n+1)}{6}, \quad S_3(n) = \frac{n^2(n+1)^2}{4},$$
$$S_4(n) = \frac{n(n+1)(2n+1)(3n^2+3n-1)}{30}$$

求解 (S_5(n)) 即得同一封闭式

$$S_5(n) = \frac{n^2(n+1)^2(2n^2+2n-1)}{12}.$$

补充题：证明二重求和可拆为两个单和的乘积

证明： $\sum_{\substack{j\in J\\k\in K}} a_j b_k = \left(\sum_{j\in J} a_j\right)\left(\sum_{k\in K} b_k\right)$

命题（有限情形）：令 (J, K) 为有限集合，给定数列 $b_{k k\in K}$。则有

$$\sum_{j\in J}\sum_{k\in K} a_j b_k = \left(\sum_{j\in J} a_j\right)\left(\sum_{k\in K} b_k\right).$$

证明：由加法对乘法的分配律与交换律，

$$\left(\sum_{j\in J} a_j\right)\left(\sum_{k\in K} b_k\right) = \sum_{j\in J}\left(a_j \sum_{k\in K} b_k\right) = \sum_{j\in J}\sum_{k\in K} a_j b_k.$$

证毕。

备注（可数无限情形）：若两列绝对可和，即$sum_{j\in J} \mid a_j \mid < \infty$ 且 $sum_{k\in K} \mid b_k \mid < \infty$，则上式仍成立，可由 Tonelli/Fubini 定理或绝对收敛下的求和交换律推出。