

数学技巧

Claris

Hangzhou Dianzi University

2019 年 1 月 12 日

Anna and Lucky Tickets

给定 n , 请统计有多少 n 位十进制数 (可以有前导 0) , 满足它是个回文串且奇数位的数位和等于偶数位的数位和。

Anna and Lucky Tickets

给定 n , 请统计有多少 n 位十进制数 (可以有前导 0) , 满足它是个回文串且奇数位的数位和等于偶数位的数位和。

- $n \leq 1000000$ 。

Anna and Lucky Tickets

给定 n , 请统计有多少 n 位十进制数 (可以有前导 0) , 满足它是个回文串且奇数位的数位和等于偶数位的数位和。

- $n \leq 1000000$ 。
- Source : XVIII Open Cup GP of Ural

Solution

- 若 n 是偶数，那么因为回文的性质，左半部分在右半部分奇偶性反转，故奇数位和必定等于偶数位和，答案为 $10^{\frac{n}{2}}$ 。

Solution

- 若 n 是偶数，那么因为回文的性质，左半部分在右半部分奇偶性反转，故奇数位和必定等于偶数位和，答案为 $10^{\frac{n}{2}}$ 。
- 考虑 n 是奇数的情况，枚举中间的数，那么剩下的数里奇数位要填 $[0, 9]$ ，偶数位要填 $[-9, 0]$ 。

Solution

- 若 n 是偶数，那么因为回文的性质，左半部分在右半部分奇偶性反转，故奇数位和必定等于偶数位和，答案为 $10^{\frac{n}{2}}$ 。
- 考虑 n 是奇数的情况，枚举中间的数，那么剩下的数里奇数位要填 $[0, 9]$ ，偶数位要填 $[-9, 0]$ 。
- 将偶数位都加上 9，变成统计 $\sum_{i=1}^n x_i = m, 0 \leq x_i \leq 9$ 的解数。

Solution

- 若 n 是偶数，那么因为回文的性质，左半部分在右半部分奇偶性反转，故奇数位和必定等于偶数位和，答案为 $10^{\frac{n}{2}}$ 。
- 考虑 n 是奇数的情况，枚举中间的数，那么剩下的数里奇数位要填 $[0, 9]$ ，偶数位要填 $[-9, 0]$ 。
- 将偶数位都加上 9，变成统计 $\sum_{i=1}^n x_i = m, 0 \leq x_i \leq 9$ 的解数。
- 容斥，枚举有 k 个 $x > 9$ ，剩下的不管，将 m 减去 $10k$ 后用隔板法计算方案数，乘以容斥系数 $(-1)^k C(n, k)$ 。

Solution

- 若 n 是偶数，那么因为回文的性质，左半部分在右半部分奇偶性反转，故奇数位和必定等于偶数位和，答案为 $10^{\frac{n}{2}}$ 。
- 考虑 n 是奇数的情况，枚举中间的数，那么剩下的数里奇数位要填 $[0, 9]$ ，偶数位要填 $[-9, 0]$ 。
- 将偶数位都加上 9，变成统计 $\sum_{i=1}^n x_i = m, 0 \leq x_i \leq 9$ 的解数。
- 容斥，枚举有 k 个 $x > 9$ ，剩下的不管，将 m 减去 $10k$ 后用隔板法计算方案数，乘以容斥系数 $(-1)^k C(n, k)$ 。
- 时间复杂度 $O(n)$ 。

Xavier is Learning to Count

给定 m 个互不相同的正整数 $a_1, a_2, \dots, a_m$ 和一个参数 p 。

Xavier is Learning to Count

给定 m 个互不相同的正整数 $a_1, a_2, \dots, a_m$ 和一个参数 p 。

请对于每个 $1 \leq k \leq p \times \max(a_i)$ ，求从 m 个数中不重复地选 p 个数相加和为 k 的方案数。

Xavier is Learning to Count

给定 m 个互不相同的正整数 $a_1, a_2, \dots, a_m$ 和一个参数 p 。

请对于每个 $1 \leq k \leq p \times \max(a_i)$ ，求从 m 个数中不重复地选 p 个数相加和为 k 的方案数。

- $m, a_i \leq 13000$ 。

Xavier is Learning to Count

给定 m 个互不相同的正整数 $a_1, a_2, \dots, a_m$ 和一个参数 p 。

请对于每个 $1 \leq k \leq p \times \max(a_i)$ ，求从 m 个数中不重复地选 p 个数相加和为 k 的方案数。

- $m, a_i \leq 13000$ 。
- $p \leq 5$ 。

Xavier is Learning to Count

给定 m 个互不相同的正整数 $a_1, a_2, \dots, a_m$ 和一个参数 p 。

请对于每个 $1 \leq k \leq p \times \max(a_i)$ ，求从 m 个数中不重复地选 p 个数相加和为 k 的方案数。

- $m, a_i \leq 13000$ 。
- $p \leq 5$ 。
- Source : BZOJ 2498

Solution

- 设这 p 个数分别为 $x_1, x_2, \dots, x_p$, 只需要让它们互不相同 , 最后答案再除以 $p!$ 即可。

Solution

- 设这 p 个数分别为 $x_1, x_2, \dots, x_p$, 只需要让它们互不相同 , 最后答案再除以 $p!$ 即可。
- 容斥 , $O(\text{Bell}(p))$ 枚举 p 个点的所有划分情况 , 钦定每个等价类里的数必须相同 , 不同等价类的数可以相同也可以不同。

Solution

- 设这 p 个数分别为 $x_1, x_2, \dots, x_p$, 只需要让它们互不相同 , 最后答案再除以 $p!$ 即可。
- 容斥 , $O(\text{Bell}(p))$ 枚举 p 个点的所有划分情况 , 钦定每个等价类里的数必须相同 , 不同等价类的数可以相同也可以不同。
- 每个等价类构造多项式 , 不同等价类之间 FFT 计算多项式乘法。

Solution

- 设这 p 个数分别为 $x_1, x_2, \dots, x_p$, 只需要让它们互不相同 , 最后答案再除以 $p!$ 即可。
- 容斥 , $O(Bell(p))$ 枚举 p 个点的所有划分情况 , 钦定每个等价类里的数必须相同 , 不同等价类的数可以相同也可以不同。
- 每个等价类构造多项式 , 不同等价类之间 FFT 计算多项式乘法。
- 假设有 k 个等价类 , 大小分别为 $size_1, size_2, \dots, size_k$, 则容斥系数为 $(-1)^{p-k} \prod_{i=1}^k (size_i - 1)!$ 。

Solution

- 设这 p 个数分别为 $x_1, x_2, \dots, x_p$, 只需要让它们互不相同 , 最后答案再除以 $p!$ 即可。
- 容斥 , $O(\text{Bell}(p))$ 枚举 p 个点的所有划分情况 , 钦定每个等价类里的数必须相同 , 不同等价类的数可以相同也可以不同。
- 每个等价类构造多项式 , 不同等价类之间 FFT 计算多项式乘法。
- 假设有 k 个等价类 , 大小分别为 $\text{size}_1, \text{size}_2, \dots, \text{size}_k$, 则容斥系数为 $(-1)^{p-k} \prod_{i=1}^k (\text{size}_i - 1)!$ 。
- 时间复杂度 $O(\text{Bell}(p)pn \log(pn))$ 。

Lines

$n \times n$ 的黑白矩阵，若任意一行、任意一列、主对角线和反对角线都满足上面的格子包含黑白两种颜色，则称这个矩阵是好的。

Lines

$n \times n$ 的黑白矩阵，若任意一行、任意一列、主对角线和反对角线都满足上面的格子包含黑白两种颜色，则称这个矩阵是好的。

求有多少好的矩阵，模质数。

Lines

$n \times n$ 的黑白矩阵，若任意一行、任意一列、主对角线和反对角线都满足上面的格子包含黑白两种颜色，则称这个矩阵是好的。

求有多少好的矩阵，模质数。

- $n \leq 300$ 。

Lines

$n \times n$ 的黑白矩阵，若任意一行、任意一列、主对角线和反对角线都满足上面的格子包含黑白两种颜色，则称这个矩阵是好的。

求有多少好的矩阵，模质数。

- $n \leq 300$ 。
- Source : AMPPZ 2018

Solution

- 容斥，等价于选择一些行/列/对角线，钦定上面的颜色都相同，剩下的不管。

Solution

- 容斥，等价于选择一些行/列/对角线，钦定上面的颜色都相同，剩下的不管。
- 设被钦定打破的限制数量为 A ，未被这些地方覆盖的格子数为 B ，这些地方形成的连通块个数为 C ，则对答案的贡献为 $(-1)^A 2^B 2^C$ 。

Solution

- 容斥，等价于选择一些行/列/对角线，钦定上面的颜色都相同，剩下的不管。
- 设被钦定打破的限制数量为 A ，未被这些地方覆盖的格子数为 B ，这些地方形成的连通块个数为 C ，则对答案的贡献为 $(-1)^A 2^B 2^C$ 。
- 对于选择了 0 条对角线的情况，是经典问题，枚举选了 i 行 j 列，则 $A = i + j$ ， $B = (n - i)(n - j)$ ，当 i, j 都不是 0 时 $C = 1$ ，否则 $C = i + j$ 。

Solution

- 容斥，等价于选择一些行/列/对角线，钦定上面的颜色都相同，剩下的不管。
- 设被钦定打破的限制数量为 A ，未被这些地方覆盖的格子数为 B ，这些地方形成的连通块个数为 C ，则对答案的贡献为 $(-1)^A 2^B 2^C$ 。
- 对于选择了 0 条对角线的情况，是经典问题，枚举选了 i 行 j 列，则 $A = i + j$ ， $B = (n - i)(n - j)$ ，当 i, j 都不是 0 时 $C = 1$ ，否则 $C = i + j$ 。
- 乘以组合数 $C(n, i)C(n, j)$ ，时间复杂度 $O(n^2)$ 。

Solution

- 对于选择了 1 条对角线的情况，可以发现 C 一定为 1。

Solution

- 对于选择了 1 条对角线的情况，可以发现 C 一定为 1。
- 假设选择了 x 行 y 列，那么未被覆盖的格子数就是 $(n - x)(n - y)$ 减去对角线上未被覆盖的格子数。

Solution

- 对于选择了 1 条对角线的情况，可以发现 C 一定为 1。
- 假设选择了 x 行 y 列，那么未被覆盖的格子数就是 $(n-x)(n-y)$ 减去对角线上未被覆盖的格子数。
- 考虑 DP，设 $f[i][j][k]$ 表示 $i \times i$ 的棋盘，选了 j 行 k 列时，所有方案的 $\frac{(-1)^A}{2^t}$ 之和，其中 t 表示对角线上未被覆盖的格子数。

Solution

- 对于选择了 1 条对角线的情况，可以发现 C 一定为 1。
- 假设选择了 x 行 y 列，那么未被覆盖的格子数就是 $(n-x)(n-y)$ 减去对角线上未被覆盖的格子数。
- 考虑 DP，设 $f[i][j][k]$ 表示 $i \times i$ 的棋盘，选了 j 行 k 列时，所有方案的 $\frac{(-1)^A}{2^t}$ 之和，其中 t 表示对角线上未被覆盖的格子数。
- 枚举第 $i+1$ 行和第 $i+1$ 列的选法转移。

Solution

- 对于选择了 1 条对角线的情况，可以发现 C 一定为 1。
- 假设选择了 x 行 y 列，那么未被覆盖的格子数就是 $(n-x)(n-y)$ 减去对角线上未被覆盖的格子数。
- 考虑 DP，设 $f[i][j][k]$ 表示 $i \times i$ 的棋盘，选了 j 行 k 列时，所有方案的 $\frac{(-1)^A}{2^t}$ 之和，其中 t 表示对角线上未被覆盖的格子数。
- 枚举第 $i+1$ 行和第 $i+1$ 列的选法转移。
- 时间复杂度 $O(n^3)$ 。

Solution

- 对于选择了 2 条对角线的情况，和上述方法类似。

Solution

- 对于选择了 2 条对角线的情况，和上述方法类似。
- 设 $f[i][j][k]$ 表示 $i \times i$ 的棋盘，选了 j 行 k 列时，所有方案的 $\frac{(-1)^A}{2^t}$ 之和，其中 t 表示对角线上未被覆盖的格子数。

Solution

- 对于选择了 2 条对角线的情况，和上述方法类似。
- 设 $f[i][j][k]$ 表示 $i \times i$ 的棋盘，选了 j 行 k 列时，所有方案的 $\frac{(-1)^A}{2^t}$ 之和，其中 t 表示对角线上未被覆盖的格子数。
- 往外扩一圈转移，从 $f[i][j][k]$ 转移到 $f[i+2][j][k]$ 。

Solution

- 对于选择了 2 条对角线的情况，和上述方法类似。
- 设 $f[i][j][k]$ 表示 $i \times i$ 的棋盘，选了 j 行 k 列时，所有方案的 $\frac{(-1)^A}{2^t}$ 之和，其中 t 表示对角线上未被覆盖的格子数。
- 往外扩一圈转移，从 $f[i][][[]]$ 转移到 $f[i+2][][[]]$ 。
- 时间复杂度 $O(n^3)$ 。

Solution

- 对于选择了 2 条对角线的情况，和上述方法类似。
- 设 $f[i][j][k]$ 表示 $i \times i$ 的棋盘，选了 j 行 k 列时，所有方案的 $\frac{(-1)^A}{2^t}$ 之和，其中 t 表示对角线上未被覆盖的格子数。
- 往外扩一圈转移，从 $f[i][j][k]$ 转移到 $f[i+2][j][k]$ 。
- 时间复杂度 $O(n^3)$ 。
- 注意当 n 是偶数且 $j = k = 0$ 时，连通块个数 $C = 2$ ，需要特判。

Solution

- 还有最后一个问题，当 2 不存在逆元的时候怎么办？

Solution

- 还有最后一个问题，当 2 不存在逆元的时候怎么办？
- 这种情况下模数 P 只能是 2。

Solution

- 还有最后一个问题，当 2 不存在逆元的时候怎么办？
- 这种情况下模数 P 只能是 2。
- 因为黑白可以交换，所以答案肯定是偶数，直接输出 0 就好了。

Perfect Square

给定正整数 n ，判断它是否是完全平方数。

Perfect Square

给定正整数 n ，判断它是否是完全平方数。

- $n \leq 10^{1000000}$ 。

Perfect Square

给定正整数 n , 判断它是否是完全平方数。

- $n \leq 10^{1000000}$ 。
- Source : XVIII Open Cup GP of Peterhof

Solution

- 问题的关键：这是个 Yes or No 的问题。

Solution

- 问题的关键：这是个 Yes or No 的问题。
- 随机选择 k 个大质数 P ，判断模 P 的二次剩余是否有解。

Solution

- 问题的关键：这是个 Yes or No 的问题。
- 随机选择 k 个大质数 P ，判断模 P 的二次剩余是否有解。
- 若质数 $P > 2$ 且 $n^{\frac{P-1}{2}} \bmod P = 1$ ，则说明有解。

Solution

- 问题的关键：这是个 Yes or No 的问题。
- 随机选择 k 个大质数 P ，判断模 P 的二次剩余是否有解。
- 若质数 $P > 2$ 且 $n^{\frac{P-1}{2}} \bmod P = 1$ ，则说明有解。
- 时间复杂度 $O(k \log P + k|n|)$ 。

Tree Distance

给定一个 n 个点的完全无向图，每条边为黑色或者白色。

Tree Distance

给定一个 n 个点的完全无向图，每条边为黑色或者白色。

请对于每个 $0 \leq k < n$ ，统计有多少生成树恰好包含 k 条黑边。

Tree Distance

给定一个 n 个点的完全无向图，每条边为黑色或者白色。

请对于每个 $0 \leq k < n$ ，统计有多少生成树恰好包含 k 条黑边。

- $n \leq 50$ 。

Solution

- 假设 k 条黑边有 ans_k 个, 那么答案可以用生成函数 $F(x) = \sum_{i=0}^{n-1} ans_i x^i$ 来表示。

Solution

- 假设 k 条黑边有 ans_k 个，那么答案可以用生成函数 $F(x) = \sum_{i=0}^{n-1} ans_i x^i$ 来表示。
- 这是个度数为 n 的多项式，只要取 n 个 x 算出 $F(x)$ 的值就可以通过拉格朗日插值在 $O(n^2)$ 的时间内得到所有系数 ans_i 。

Solution

- 假设 k 条黑边有 ans_k 个，那么答案可以用生成函数 $F(x) = \sum_{i=0}^{n-1} ans_i x^i$ 来表示。
- 这是个度数为 n 的多项式，只要取 n 个 x 算出 $F(x)$ 的值就可以通过拉格朗日插值在 $O(n^2)$ 的时间内得到所有系数 ans_i 。
- 假设 $x = w$ ，那么对于一棵 k 条黑边的生成树，我们希望给函数贡献 w^k 。

Solution

- 假设 k 条黑边有 ans_k 个，那么答案可以用生成函数 $F(x) = \sum_{i=0}^{n-1} ans_i x^i$ 来表示。
- 这是个度数为 n 的多项式，只要取 n 个 x 算出 $F(x)$ 的值就可以通过拉格朗日插值在 $O(n^2)$ 的时间内得到所有系数 ans_i 。
- 假设 $x = w$ ，那么对于一棵 k 条黑边的生成树，我们希望给函数贡献 w^k 。
- 构造新图：白边不变，黑边复制到 k 份。那么新图的生成树个数就是 $F(w)$ 的值。

Solution

- 假设 k 条黑边有 ans_k 个，那么答案可以用生成函数 $F(x) = \sum_{i=0}^{n-1} ans_i x^i$ 来表示。
- 这是个度数为 n 的多项式，只要取 n 个 x 算出 $F(x)$ 的值就可以通过拉格朗日插值在 $O(n^2)$ 的时间内得到所有系数 ans_i 。
- 假设 $x = w$ ，那么对于一棵 k 条黑边的生成树，我们希望给函数贡献 w^k 。
- 构造新图：白边不变，黑边复制到 k 份。那么新图的生成树个数就是 $F(w)$ 的值。
- 利用 Matrix-Tree 定理计算行列式即可。

Solution

- 假设 k 条黑边有 ans_k 个，那么答案可以用生成函数 $F(x) = \sum_{i=0}^{n-1} ans_i x^i$ 来表示。
- 这是个度数为 n 的多项式，只要取 n 个 x 算出 $F(x)$ 的值就可以通过拉格朗日插值在 $O(n^2)$ 的时间内得到所有系数 ans_i 。
- 假设 $x = w$ ，那么对于一棵 k 条黑边的生成树，我们希望给函数贡献 w^k 。
- 构造新图：白边不变，黑边复制到 k 份。那么新图的生成树个数就是 $F(w)$ 的值。
- 利用 Matrix-Tree 定理计算行列式即可。
- 时间复杂度 $O(n^4)$ 。

Counting Sheep in Ami Dongsuo

给定一个 n 个点, m 条边的 DAG, 每个点点权 a_i 是 $[1, w]$ 之间的整数。

Counting Sheep in Ami Dongsuo

给定一个 n 个点, m 条边的 DAG, 每个点点权 a_i 是 $[1, w]$ 之间的整数。

三个人从 S 出发, 分别走若干步后到达 A, B, C (步数可以不同), 则会对 $ans[a_A + a_B + a_C]$ 贡献 1。

Counting Sheep in Ami Dongsuo

给定一个 n 个点, m 条边的 DAG, 每个点点权 a_i 是 $[1, w]$ 之间的整数。

三个人从 S 出发, 分别走若干步后到达 A, B, C (步数可以不同), 则会对 $ans[a_A + a_B + a_C]$ 贡献 1。

请枚举所有的 S, A, B, C 以及所有不同的路线, 输出所有 $ans[]$ 的值。

Counting Sheep in Ami Dongsuo

给定一个 n 个点, m 条边的 DAG, 每个点点权 a_i 是 $[1, w]$ 之间的整数。

三个人从 S 出发, 分别走若干步后到达 A, B, C (步数可以不同), 则会对 $ans[a_A + a_B + a_C]$ 贡献 1。

请枚举所有的 S, A, B, C 以及所有不同的路线, 输出所有 $ans[]$ 的值。

- $n \leq 10000, m \leq 30000$ 。

Counting Sheep in Ami Dongsuo

给定一个 n 个点, m 条边的 DAG, 每个点点权 a_i 是 $[1, w]$ 之间的整数。

三个人从 S 出发, 分别走若干步后到达 A, B, C (步数可以不同), 则会对 $ans[a_A + a_B + a_C]$ 贡献 1。

请枚举所有的 S, A, B, C 以及所有不同的路线, 输出所有 $ans[]$ 的值。

- $n \leq 10000, m \leq 30000$ 。
- $w \leq 400$ 。

Counting Sheep in Ami Dongsuo

给定一个 n 个点, m 条边的 DAG, 每个点点权 a_i 是 $[1, w]$ 之间的整数。

三个人从 S 出发, 分别走若干步后到达 A, B, C (步数可以不同), 则会对 $ans[a_A + a_B + a_C]$ 贡献 1。

请枚举所有的 S, A, B, C 以及所有不同的路线, 输出所有 $ans[]$ 的值。

- $n \leq 10000, m \leq 30000$ 。
- $w \leq 400$ 。
- Source : 2018-2019 ICPC, Asia Shenyang Regional Contest

Solution

- 设 $f[i][j][k]$ 表示从 i 出发走若干步到达 j 个终点，终点的点权和为 k 的方案数。

Solution

- 设 $f[i][j][k]$ 表示从 i 出发走若干步到达 j 个终点，终点的点权和为 k 的方案数。
- 转移显然，是个卷积的形式。

Solution

- 设 $f[i][j][k]$ 表示从 i 出发走若干步到达 j 个终点，终点的点权和为 k 的方案数。
- 转移显然，是个卷积的形式。
- FFT 优化，时间复杂度 $O(nw \log w)$ 。

Solution

- 设 $f[i][j][k]$ 表示从 i 出发走若干步到达 j 个终点，终点的点权和为 k 的方案数。
- 转移显然，是个卷积的形式。
- FFT 优化，时间复杂度 $O(nw \log w)$ 。
- 不幸的是模数是 $10^9 + 7$ ，而且优化后的 FFT 也不能在时限内出解。

Solution

- 将每个状态写成生成函数的形式：

$$f[i][j](x) = \sum_{k=0}^{3w} f[i][j][k]x^k。$$

Solution

- 将每个状态写成生成函数的形式：

$$f[i][j](x) = \sum_{k=0}^{3w} f[i][j][k]x^k。$$

- 同理，选取 $O(w)$ 个 x 的值，计算出答案的生成函数的值，然后插值得到每一项。

Solution

- 将每个状态写成生成函数的形式：

$$f[i][j](x) = \sum_{k=0}^{3w} f[i][j][k]x^k。$$

- 同理，选取 $O(w)$ 个 x 的值，计算出答案的生成函数的值，然后插值得到每一项。
- 去掉了 k 这一维后，转移显然是 $O(1)$ 的。

Solution

- 将每个状态写成生成函数的形式：

$$f[i][j](x) = \sum_{k=0}^{3w} f[i][j][k]x^k。$$

- 同理，选取 $O(w)$ 个 x 的值，计算出答案的生成函数的值，然后插值得到每一项。
- 去掉了 k 这一维后，转移显然是 $O(1)$ 的。
- 时间复杂度 $O(nw + w^2)$ 。

Cube Summation

给定 n , 考虑所有它的整数拆分。

Cube Summation

给定 n , 考虑所有它的整数拆分。

比如当 $n = 3$ 时 , 有 3 种拆分方法 : $1 + 1 + 1, 1 + 2, 3$ 。

Cube Summation

给定 n ，考虑所有它的整数拆分。

比如当 $n = 3$ 时，有 3 种拆分方法： $1 + 1 + 1, 1 + 2, 3$ 。

请遍历 n 的所有拆分方式，统计每种拆分方式中数字个数的立方和。比如 $n = 3$ 时答案 $3^3 + 2^3 + 1^3 = 36$ 。

Cube Summation

给定 n ，考虑所有它的整数拆分。

比如当 $n = 3$ 时，有 3 种拆分方法： $1 + 1 + 1, 1 + 2, 3$ 。

请遍历 n 的所有拆分方式，统计每种拆分方式中数字个数的立方和。比如 $n = 3$ 时答案 $3^3 + 2^3 + 1^3 = 36$ 。

■ $n \leq 100000$ 。

Cube Summation

给定 n , 考虑所有它的整数拆分。

比如当 $n = 3$ 时 , 有 3 种拆分方法 : $1 + 1 + 1, 1 + 2, 3$ 。

请遍历 n 的所有拆分方式 , 统计每种拆分方式中数字个数的立方和。比如 $n = 3$ 时答案 $3^3 + 2^3 + 1^3 = 36$ 。

- $n \leq 100000$ 。
- Source : 2017 Multi-University Training Contest - Team 10

Solution

- 设 $f[i][j]$ 表示用 $[1, i]$ 拼出 j 的方案数, 则
$$f[i][j] = f[i][j - i] + f[i - 1][j],$$
表示多要一个 i 或者不要 i 了。

Solution

- 设 $f[i][j]$ 表示用 $[1, i]$ 拼出 j 的方案数, 则
$$f[i][j] = f[i][j - i] + f[i - 1][j],$$
表示多要一个 i 或者不要 i 了。
- 其实就是完全背包。

Solution

- 设 $f[i][j]$ 表示用 $[1, i]$ 拼出 j 的方案数, 则
$$f[i][j] = f[i][j - i] + f[i - 1][j],$$
表示多要一个 i 或者不要 i 了。
- 其实就是完全背包。
- 直接 DP 是 $O(n^2)$ 的, 不能承受。

Solution

- 设 $lim = \sqrt{n}$, 注意到一个数如果 $> lim$, 那么它最多只会用 lim 个。

Solution

- 设 $lim = \sqrt{n}$, 注意到一个数如果 $> lim$, 那么它最多只会用 lim 个。
- 用上述方法 DP 出 $f[lim][]$ 后 , 设 $g[i][j]$ 表示用 i 个 $> lim$ 的数拼出 j 的方案数。

Solution

- 设 $lim = \sqrt{n}$, 注意到一个数如果 $> lim$, 那么它最多只会用 lim 个。
- 用上述方法 DP 出 $f[lim][j]$ 后 , 设 $g[i][j]$ 表示用 i 个 $> lim$ 的数拼出 j 的方案数。
- 初始值 $g[0][j] = f[lim][j]$ 。

Solution

- 设 $lim = \sqrt{n}$, 注意到一个数如果 $> lim$, 那么它最多只会用 lim 个。
- 用上述方法 DP 出 $f[lim][j]$ 后 , 设 $g[i][j]$ 表示用 i 个 $> lim$ 的数拼出 j 的方案数。
- 初始值 $g[0][j] = f[lim][j]$ 。
- 转移 : $g[i][j] = g[i-1][j - lim - 1] + g[i][j - i]$, 表示要么新加一个 $lim + 1$, 要么把 $> lim$ 的所有数都加 1。

Solution

- 设 $lim = \sqrt{n}$, 注意到一个数如果 $> lim$, 那么它最多只会用 lim 个。
- 用上述方法 DP 出 $f[lim][j]$ 后 , 设 $g[i][j]$ 表示用 i 个 $> lim$ 的数拼出 j 的方案数。
- 初始值 $g[0][j] = f[lim][j]$ 。
- 转移 : $g[i][j] = g[i-1][j - lim - 1] + g[i][j - i]$, 表示要么新加一个 $lim + 1$, 要么把 $> lim$ 的所有数都加 1。
- 时间复杂度 $O(n\sqrt{n})$ 。

Solution

- 等等，这题不是求立方和吗？

Solution

- 等等，这题不是求立方和吗？
-

$$x^k = \sum_{i=1}^k Stirling2(k, i) i! C(x, i)$$

Solution

- 等等，这题不是求立方和吗？
-

$$x^k = \sum_{i=1}^k \text{Stirling2}(k, i) i! C(x, i)$$

- 也就是说设 t 表示一个拆分方案里的数字个数，我们只要知道所有方案的 $C(t, i)$ 的和就可以了。

Solution

- 等等，这题不是求立方和吗？
-

$$x^k = \sum_{i=1}^k \text{Stirling2}(k, i) i! C(x, i)$$

- 也就是说设 t 表示一个拆分方案里的数字个数，我们只要知道所有方案的 $C(t, i)$ 的和就可以了。
- 而根据杨辉三角 $C(t, i) = C(t-1, i-1) + C(t-1, i)$ ，所以配合整数拆分的 DP 方式，每次新加数字的时候跟着转移 $C(t, i)$ 就好了。

Product of Roots

有两个非负整数数列 $a_1, a_2, \dots, a_n$ 以及 $b_1, b_2, \dots, b_m$ 。

Product of Roots

有两个非负整数数列 $a_1, a_2, \dots, a_n$ 以及 $b_1, b_2, \dots, b_m$ 。

$$f(x) = \prod_{i=1}^n (1 + a_i x) \quad g(x) = \prod_{j=1}^m (1 + b_j x)$$

$$h(x) = \prod_{i=1}^n \prod_{j=1}^m (1 + a_i b_j x)$$

Product of Roots

有两个非负整数数列 $a_1, a_2, \dots, a_n$ 以及 $b_1, b_2, \dots, b_m$ 。

$$f(x) = \prod_{i=1}^n (1 + a_i x) \quad g(x) = \prod_{j=1}^m (1 + b_j x)$$

$$h(x) = \prod_{i=1}^n \prod_{j=1}^m (1 + a_i b_j x)$$

给定 n, m 以及 f, g 的每一项的系数，求 h 的前 k 项的系数。

Product of Roots

有两个非负整数数列 $a_1, a_2, \dots, a_n$ 以及 $b_1, b_2, \dots, b_m$ 。

$$f(x) = \prod_{i=1}^n (1 + a_i x) \quad g(x) = \prod_{j=1}^m (1 + b_j x)$$

$$h(x) = \prod_{i=1}^n \prod_{j=1}^m (1 + a_i b_j x)$$

给定 n, m 以及 f, g 的每一项的系数，求 h 的前 k 项的系数。

- $n, m, k \leq 100000$ 。

Product of Roots

有两个非负整数数列 $a_1, a_2, \dots, a_n$ 以及 $b_1, b_2, \dots, b_m$ 。

$$f(x) = \prod_{i=1}^n (1 + a_i x) \quad g(x) = \prod_{j=1}^m (1 + b_j x)$$

$$h(x) = \prod_{i=1}^n \prod_{j=1}^m (1 + a_i b_j x)$$

给定 n, m 以及 f, g 的每一项的系数，求 h 的前 k 项的系数。

- $n, m, k \leq 100000$ 。
- Source : XVIII Open Cup GP of Peterhof

Solution



$$f(x) = \prod_{i=1}^n (1 + a_i x) = \exp \left(\sum_{i=1}^n \ln(1 + a_i x) \right)$$

Solution



$$f(x) = \prod_{i=1}^n (1 + a_i x) = \exp \left(\sum_{i=1}^n \ln(1 + a_i x) \right)$$

- 将 $\ln(1 + a_i x)$ 泰勒展开：

Solution



$$f(x) = \prod_{i=1}^n (1 + a_i x) = \exp \left(\sum_{i=1}^n \ln(1 + a_i x) \right)$$

- 将 $\ln(1 + a_i x)$ 泰勒展开：



$$\begin{aligned} f(x) &= \exp \left(\sum_{i=1}^n \ln(1 + a_i x) \right) = \exp \left(\sum_{i=1}^n \sum_{j \geq 1} \frac{(-1)^{j+1}}{j} x^j a_i^j \right) \\ &= \exp \left(\sum_{j \geq 1} \frac{(-1)^{j+1}}{j} x^j \sum_{i=1}^n a_i^j \right) = \exp \left(\sum_{j \geq 1} \frac{(-1)^{j+1}}{j} x^j A_j \right) \end{aligned}$$

Solution



$$\begin{aligned}g(x) &= \exp \left(\sum_{j \geq 1} \frac{(-1)^{j+1}}{j} x^j B_j \right) \\h(x) &= \exp \left(\sum_{j \geq 1} \frac{(-1)^{j+1}}{j} x^j \sum_{i=1}^n \sum_{k=1}^m a_i^j b_k^j \right) \\&= \exp \left(\sum_{j \geq 1} \frac{(-1)^{j+1}}{j} x^j \sum_{i=1}^n a_i^j \sum_{k=1}^m b_k^j \right) \\&= \exp \left(\sum_{j \geq 1} \frac{(-1)^{j+1}}{j} x^j A_j B_j \right)\end{aligned}$$

Solution

- 只要把 f 和 g 求 $\ln$ 就可以求出所有 A_j 和 B_j 。

Solution

- 只要把 f 和 g 求 $\ln$ 就可以求出所有 A_j 和 B_j 。
- 最后用多项式 $\exp$ 求出 h 即可。

Solution

- 只要把 f 和 g 求 $\ln$ 就可以求出所有 A_j 和 B_j 。
- 最后用多项式 $\exp$ 求出 h 即可。
- 时间复杂度 $O(n \log n)$ 。

Thank you!