二项式反演的四种形式、证明、相关题目

前置知识
讲解066 ~ 讲解069 - 动态规划基础内容，本节课题目涉及动态规划
讲解078 ~ 讲解079 - 树型dp，本节课题目5需要
讲解118 - 树上问题专题1，递归函数改成迭代函数，防止系统栈溢出，本节课题目5需要
讲解144 - 二项式定理

上节课讲述
二项式定理证明和相关题目

本节课讲述
二项式反演的四种形式、证明、相关题目

二项式反演的四种形式、证明、相关题目

二项式反演的四种形式

$$g(n)=\sum_{i=0}^{n}(-1)^i\binom{n}{i}f(i) \Longleftrightarrow f(n)=\sum_{i=0}^{n}(-1)^i\binom{n}{i}g(i)$$

$$g(n)=\sum_{i=0}^{n}\binom{n}{i}f(i) \Longleftrightarrow f(n)=\sum_{i=0}^{n}(-1)^{n-i}\binom{n}{i}g(i)$$

$$g(n)=\sum_{i=n}^{N}(-1)^i\binom{i}{n}f(i) \Longleftrightarrow f(n)=\sum_{i=n}^{N}(-1)^i\binom{i}{n}g(i)$$

$$g(n)=\sum_{i=n}^{N}\binom{i}{n}f(i) \Longleftrightarrow f(n)=\sum_{i=n}^{N}(-1)^{i-n}\binom{i}{n}g(i)$$

课上图解如何使用

f(i)、g(i)，可以是数列或者函数

具备反演的特征，就可以互相转化

形式二、形式四最常用

不需要用容斥来理解(徒增烦恼)

本节课会讲解完整的证明(包懂)

左程云

二项式反演的四种形式、证明、相关题目

题目1
信封问题(错排问题)
一共n个人，每个人都写了一封信
每个人必须寄出一封信，每个人必须收到一封信，并且不能自己寄给自己
返回一共有多少种寄信的方法
1 <= n <= 20
测试链接 : https://www.luogu.com.cn/problem/P1595

二项式反演的四种形式、证明、相关题目

二项式系数的组合分解公式

这是二项式反演证明过程的重要一步

$$\binom{j}{i}\binom{i}{n}=\binom{j}{n}\binom{j-n}{j-i}$$

代数证明：

从左边开始：

$$\binom{j}{i}\binom{i}{n}=\frac{j!}{i!(j-i)!}\cdot\frac{i!}{n!(i-n)!}$$

化简后为：

$$\frac{j!}{n!(i-n)!(j-i)!}$$

这个结果已经化简为 $\binom{j}{n}\cdot\frac{(j-n)!}{(j-i)!(i-n)!}$，现在我们来看右边的表达式。

右边：

$$\binom{j}{n}\binom{j-n}{j-i}=\frac{j!}{n!(j-n)!}\cdot\frac{(j-n)!}{(j-i)!(i-n)!}$$

可以看到，右边也化简为：

$$\frac{j!}{n!(i-n)!(j-i)!}$$

因此，左右两边相等，证明了这个恒等式的正确性。

左程云

二项式反演的四种形式、证明、相关题目

形式一：

$$g(n)=\sum_{i=0}^{n}(-1)^i\binom{n}{i}f(i) \iff f(n)=\sum_{i=0}^{n}(-1)^i\binom{n}{i}g(i)$$

证明：

从 $g(n)$ 的定义开始：

$$g(n)=\sum_{i=0}^{n}(-1)^i\binom{n}{i}f(i)$$

假设 $f(n)$ 可以通过 $g(i)$ 的线性组合表示：

$$f(n)=\sum_{i=0}^{n}(-1)^i\binom{n}{i}g(i)$$

将 $g(i)$ 的定义代入 $f(n)$：

$$f(n)=\sum_{i=0}^{n}(-1)^i\binom{n}{i}\sum_{j=0}^{i}(-1)^j\binom{i}{j}f(j)$$

交换求和顺序，得到：

$$f(n)=\sum_{j=0}^{n}f(j)\sum_{i=j}^{n}(-1)^{i+j}\binom{n}{i}\binom{i}{j}$$

应用二项式系数恒等式：

$$\binom{n}{i}\binom{i}{j}=\binom{n}{j}\binom{n-j}{n-i}$$

将这个恒等式代入后，$f(n)$ 化简为：

$$f(n)=\sum_{j=0}^{n}f(j)\binom{n}{j}\sum_{i=j}^{n}(-1)^{i+j}\binom{n-j}{n-i}$$

现在，求和项 $\sum_{i=j}^{n}(-1)^{i+j}\binom{n-j}{n-i}$ 是 $(1-1)^{n-j}$，即：

$$(1-1)^{n-j}=0 \quad \text{当} \quad j\neq n$$

当 $j=n$ 时，$(1-1)^0=1$。

因此，最终我们得到 Kronecker delta 函数：

$$f(n)=f(n)$$

反演公式成立。

左程云

二项式反演的四种形式、证明、相关题目

形式二：

$$g(n)=\sum_{i=0}^{n}\binom{n}{i}f(i) \iff f(n)=\sum_{i=0}^{n}(-1)^{n-i}\binom{n}{i}g(i)$$

证明：

从 $g(n)$ 的定义开始：

$$g(n)=\sum_{i=0}^{n}\binom{n}{i}f(i)$$

假设 $f(n)$ 可以通过 $g(i)$ 的线性组合表示：

$$f(n)=\sum_{i=0}^{n}(-1)^{n-i}\binom{n}{i}g(i)$$

将 $g(i)$ 的定义代入：

$$f(n)=\sum_{i=0}^{n}(-1)^{n-i}\binom{n}{i}\sum_{j=0}^{i}\binom{i}{j}f(j)$$

交换求和顺序，得到：

$$f(n)=\sum_{j=0}^{n}f(j)\sum_{i=j}^{n}(-1)^{n-i}\binom{n}{i}\binom{i}{j}$$

应用二项式系数恒等式：

$$\binom{n}{i}\binom{i}{j}=\binom{n}{j}\binom{n-j}{n-i}$$

将这个恒等式代入后，$f(n)$ 化简为：

$$f(n)=\sum_{j=0}^{n}f(j)\binom{n}{j}\sum_{i=j}^{n}(-1)^{n-i}\binom{n-j}{n-i}$$

现在，求和项 $\sum_{i=j}^{n}(-1)^{n-i}\binom{n-j}{n-i}$ 是 $(1-1)^{n-j}$，即：

$$(1-1)^{n-j}=0 \quad \text{当} \quad j\neq n$$

当 $j=n$ 时，$(1-1)^0=1$。

最终我们得到：

$$f(n)=f(n)$$

反演公式成立。

左程云

二项式反演的四种形式、证明、相关题目

形式三：

$$g(n)=\sum_{i=n}^{N}(-1)^{i}\binom{i}{n}f(i)\iff f(n)=\sum_{i=n}^{N}(-1)^{i}\binom{i}{n}g(i)$$

证明：

从 $g(n)$ 的定义开始：

$$g(n)=\sum_{i=n}^{N}(-1)^{i}\binom{i}{n}f(i)$$

假设 $f(n)$ 可以通过 $g(i)$ 的线性组合表示：

$$f(n)=\sum_{i=n}^{N}(-1)^{i}\binom{i}{n}g(i)$$

将 $g(i)$ 的定义代入 $f(n)$ 的表达式：

$$f(n)=\sum_{i=n}^{N}(-1)^{i}\binom{i}{n}\sum_{j=i}^{N}(-1)^{j}\binom{j}{i}f(j)$$

交换求和顺序，得到：

$$f(n)=\sum_{j=n}^{N}f(j)\sum_{i=n}^{j}(-1)^{i+j}\binom{i}{n}\binom{j}{i}$$

应用二项式系数恒等式：

$$\binom{j}{i}\binom{i}{n}=\binom{j}{n}\binom{j-n}{j-i}$$

将这个恒等式代入后，$f(n)$ 化简为：

$$f(n)=\sum_{j=n}^{N}f(j)\binom{j}{n}\sum_{i=n}^{j}(-1)^{i+j}\binom{j-n}{j-i}$$

现在，求和项 $\sum_{i=n}^{j}(-1)^{i+j}\binom{j-n}{j-i}$ 是 $(1-1)^{j-n}$，即：

$$(1-1)^{j-n}=0\quad 当\quad j\neq n$$

当 $j=n$ 时，$(1-1)^{0}=1$。

最终我们得到：

$$f(n)=f(n)$$

反演公式成立。

左程云

二项式反演的四种形式、证明、相关题目

形式四：

$$g(n)=\sum_{i=n}^{N}\binom{i}{n}f(i) \iff f(n)=\sum_{i=n}^{N}(-1)^{i-n}\binom{i}{n}g(i)$$

证明：

从 $g(n)$ 的定义开始：

$$g(n)=\sum_{i=n}^{N}\binom{i}{n}f(i)$$

假设 $f(n)$ 可以通过 $g(i)$ 的线性组合表示：

$$f(n)=\sum_{i=n}^{N}(-1)^{i-n}\binom{i}{n}g(i)$$

将 $g(i)$ 的定义代入 $f(n)$ 的表达式：

$$f(n)=\sum_{i=n}^{N}(-1)^{i-n}\binom{i}{n}\sum_{j=i}^{N}\binom{j}{i}f(j)$$

交换求和顺序，得到：

$$f(n)=\sum_{j=n}^{N}f(j)\sum_{i=n}^{j}(-1)^{i-n}\binom{i}{n}\binom{j}{i}$$

应用二项式系数恒等式：

$$\binom{j}{i}\binom{i}{n}=\binom{j}{n}\binom{j-n}{j-i}$$

将这个恒等式代入后，$f(n)$ 化简为：

$$f(n)=\sum_{j=n}^{N}f(j)\binom{j}{n}\sum_{i=n}^{j}(-1)^{i-n}\binom{j-n}{j-i}$$

现在，求和项 $\sum_{i=n}^{j}(-1)^{i-n}\binom{j-n}{j-i}$ 是 $(1-1)^{j-n}$，即：

$$(1-1)^{j-n}=0 \quad \text{当} \quad j\neq n$$

当 $j=n$ 时，$(1-1)^{0}=1$。

最终我们得到：

$$f(n)=f(n)$$

反演公式成立。

左程云

二项式反演的四种形式、证明、相关题目

题目2
集合计数
一共有n个不同的数，能构成2^n个不同集合
在2^n个集合中挑出若干个集合，至少挑一个
希望这若干个集合的交集，正好有k个数
返回挑选集合的方案数，答案对 1000000007 取余
1 <= n <= 10^6
0 <= k <= n
测试链接 : https://www.luogu.com.cn/problem/P10596

二项式反演形式四

(钦定k个且至少 -> 恰好k个)的转化

二项式反演的四种形式、证明、相关题目

题目3

分特产

一共有m种特产，arr[i]表示i种特产有几个，同一种特产的物品之间无差别

一共有n个同学，同学之间认为有差别，每个同学至少要得到一个特产

返回分配特产的方法数，答案对 1000000007 取余

0 <= n、m <= 1000

0 <= arr[i] <= 1000

测试链接 : https://www.luogu.com.cn/problem/P5505

强烈建议先看懂题目2

二项式反演形式四

(钦定k个且至少 -> 恰好k个)的转化

隔板法分析m个相同物品分给n个人的方法数

二项式反演的四种形式、证明、相关题目

题目4
已经没有什么好害怕的了
给定两个长度为n的数组，a[i]表示第i个糖果的能量，b[i]表示第i个药片的能量
所有能量数值都不相同，每一个糖果要选一个药片进行配对
如果配对之后，糖果能量 > 药片能量，称为糖果大的配对
如果配对之后，糖果能量 < 药片能量，称为药片大的配对
希望做到，糖果大的配对数量 = 药片大的配对数量 + k，返回配对方法数，答案对 1000000009 取余
举例，a = [5, 35, 15, 45]，b = [40, 20, 10, 30]，k = 2，返回4，因为有4种配对方法
(5-40，35-20，15-10，45-30)、(5-40，35-30，15-10，45-20)
(5-20，35-30，15-10，45-40)、(5-30，35-20，15-10，45-40)
1 <= n <= 2000   0 <= k <= n
测试链接 : https://www.luogu.com.cn/problem/P4859

二项式反演形式四 + (钦定k个且至少 -> 恰好k个)的转化 + 背包dp

左程云

二项式反演的四种形式、证明、相关题目

题目5
游戏
一共有n个节点，n <= 5000，n为偶数，其中有m个点属于小A，有m个点属于小B，m为n的一半
给定n-1条边，节点之间组成一颗树，1号节点是根节点
给定长度为n的数组arr，arr[i]的值表示i号节点由谁拥有，0为小A拥有，1为小B拥有
游戏有m回合，每回合都有胜负，两人需要选择一个自己拥有、但之前没选过的点，作为本回合当前点
小A当前点的子树里有小B当前点，则小A胜；小B当前点的子树里有小A当前点，则小B胜；否则平局
返回m回合里能出现k次非平局的游戏方法数，打印k=0..m时的所有答案，对 998244353 取余
两场游戏视为不同的定义：当且仅当存在小A拥有的点x，小B在小A选择x的那个回合所选择的点不同
测试链接 : https://www.luogu.com.cn/problem/P6478

二项式反演形式四 + (钦定k个且至少 -> 恰好k个)的转化

树型dp + 树型dp的枚举行为利用子树的节点数做上限进行复杂度优化 + 递归函数改成迭代函数