Комбинаторные вычисления требуют предварительного анализа и количественной оценки исходных задач и используемых алгоритмов. Задачи обычно оцениваются с точки зрения размера, то есть общего количества различных вариантов, среди которых нужно найти решение, а алгоритмы оцениваются с точки зрения сложности. При этом различают сложность по времени (или временную сложность), то есть количество необходимых шагов алгоритма, и сложность по памяти (или ёмкостную сложность), то есть объем памяти, необходимый для работы алгоритма.

Комбинаторные конфигурации

Во многих практических случаях возникает необходимость подсчитать количество возможных комбинаций объектов, удовлетворяющих определенным условиям. Такие задачи называются комбинаторными. Разнообразие комбинаторных задач не поддается исчерпывающему описанию, но среди них есть целый ряд особенно часто встречающихся, для которых известны способы подсчета.

Комбинаторные задачи

Для формулировки и решения комбинаторных задач используются различные модели комбинаторных конфигураций. Рассмотрим следующие две наиболее популярные.

1. Дано n предметов. Их нужно разместить по m ящикам так, чтобы выполнялись заданные ограничения. Сколькими способами это можно сделать?
2. Рассмотрим множество функций

$$F\colon X \to Y, \quad \text{где } |X| = n,\ |Y| = m,\ X = \{1, \ldots, n\}.$$

Не ограничивая общности, можно считать, что

$$Y = \{1, \ldots, m\},\ F = \langle F(1), \ldots, F(n) \rangle,\ 1 \leqslant F(i) \leqslant m.$$

Сколько существует функций F, удовлетворяющих заданным ограничениям?

Большей частью соответствие конфигураций, описанных на «языке ящиков» и на «языке функций», очевидно, поэтому доказательство правильности способа подсчета (вывод формулы) можно провести на любом языке. Если сведение одной модели к другой не очевидно, то оно включается в доказательство.

Размещения

Число всех функций (при отсутствии ограничений), или число всех возможных способов разместить n предметов по m ящикам называется, числом размещений и обозначается $U(m,n)$.

Теорема

$$U(m, n) = m^n.$$

Доказательство
Каждый из n предметов можно разместить m способами.

Пример
При игре в кости бросаются две кости и выпавшие на верхних гранях очки складываются. Какова вероятность выбросить 12 очков? Каждый возможный исход соответствует функции $F: \{1,2\} \to \{1,2,3,4,5,6\}$ (аргумент — номер кости, результат — очки на верхней грани). Таким образом, всего возможно $U(6,2) = 6^2 = 36$ различных исходов. Из них только один $(6 + 6)$ дает двенадцать очков. Вероятность $1/36$.

Размещения без повторений
Число инъективных функций, или число всех возможных способов разместить n предметов по m ящикам, не более чем по одному в ящик, называется числом размещений без повторений и обозначается $A(m, n)$ или $[m]_n$, или $(m)_n$.

Теорема

$$A(m,n) = \frac{m!}{(m-n)!}$$

Доказательство
Ящик для первого предмета можно выбрать m способами, для второго - $(m - 1)$ способами, и т. д. Таким образом,

$$A(m,n) = m \cdot (m-1) \cdot \ldots \cdot (m-n+1) = \frac{m!}{(m-n)!}.$$

По определению считают, что $A(m, n):=0$ при $n>m$ и $A(m, 0):=1$.

Пример
В некоторых видах спортивных соревнований исходом является определение участников, занявших первое, второе и третье места. Сколько возможно различных исходов, если в соревновании участвуют n участников? Каждый возможный исход соответствует функции $F: \{1,2,3\} \to \{1,.n\}$ (аргумент — номер призового места, результат — номер участника). Таким образом, всего возможно $A(n, 3) = n(n - 1)(n - 2)$ различных исходов.

Перестановки
Число взаимнооднозначных функций, или число перестановок n предметов, обозначается $P(n)$.

Теорема

$$P(n) = n!$$

Доказательство

$$P(n) = A(n,n) = n \cdot (n-1) \cdot \cdots \cdot (n-n+1) = n \cdot (n-1) \cdot \ldots \cdot 1 = n!.$$

Пример

Последовательность $\mathcal{E} = (E_1, \ldots, E_m)$ непустых подмножеств множества E ($\mathcal{E} \subset 2^E$, $E_i \subset E$, $E_i \neq \varnothing$) называется цепочкой в E, если

$$\forall i \in 1..m-1 \; E_i \subset E_{i+1} \,\&\, E_i \neq E_{i+1}.$$

Цепочка E называется полной цепочкой в E, если $|\mathcal{E}| = |E|$. Сколько существует полных цепочек? Очевидно, что в полной цепочке каждое следующее подмножество E_{i+1} получено из предыдущего подмножества E_i добавлением ровно одного элемента из E и, таким образом,

$$|E_1| = 1,\, |E_2| = 2,\, \ldots,\, |E_m| = |E| = m.$$

Следовательно, полная цепочка определяется порядком, в котором элементы E добавляются для образования очередного элемента полной цепочки. Отсюда количество полных цепочек — это количество перестановок элементов множества E, равное $m!$.

Сочетания

Число строго монотонных функций, или число размещений n неразличимых предметов по m ящикам, не более чем по одному в ящик, то есть число способов выбрать из m ящиков n ящиков с предметами, называется числом сочетаний и обозначается $C(m,n)$ или C_m^n или $\binom{n}{m}$.

Теорема

$$C(m,n) = \frac{m!}{n!(m-n)!}$$

Доказательство

1. Число размещений без повторений нужно разделить на число перестановок, поскольку предметы неразличимы.
2. Число сочетаний является числом строго монотонных функций, потому что строго монотонная функция $F\colon 1..n \to 1..m$ определяется набором своих значений, причем $1 \leqslant F(1) < \cdots < F(n) \leqslant m$. Другими словами, каждая строго монотонная функция определяется выбором n чисел из диапазона $1..m$. Таким образом, число строго монотонных функций равно числу n-элементных подмножеств m-элементного множества, которое, в свою очередь, равно числу способов выбрать n ящиков с предметами из m ящиков.

По определению $C(m,n) := 0$ при $n > m$.

Пример

В начале игры в домино каждому играющему выдается 7 костей из имеющихся 28 различных костей. Сколько существует различных комбинаций костей, которые игрок может получить в начале игры? Очевидно, что искомое число равно числу 7-элементных подмножеств 28-элементного множества. Имеем:

$$C(28,7) = \frac{28!}{7!(28-7)!} = \frac{28 \cdot 27 \cdot 26 \cdot 25 \cdot 24 \cdot 23 \cdot 22}{7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1} = 1184040.$$

Сочетания с повторениями

Число монотонных функций, или число размещений n неразличимых предметов по m ящикам, называется числом сочетаний с повторениями и обозначается

$$V(m, n).$$

Теорема

$$V(m, n) = C(n + m - 1, n).$$

Доказательство

Монотонной функции $f\colon \{1, \ldots, n\} \to \{1, \ldots, m\}$ однозначно соответствует строго монотонная функция $f'\colon \{1, \ldots, n\} \to \{1, \ldots, n + m - 1\}$. Это соответствие устанавливается следующей процедурой.

$f'(1) := f(1)$ { первые элементы совпадают }
for i **from** 2 **to** n **do**
  if $f(i) = f(i-1)$ **then**
    $f'(i) := f'(i-1) + 1$ { плато }
  else
    $f'(i) := f'(i-1) + f(i) - f(i-1)$ { подъем }
  end if
end for

Пример

Сколькими способами можно рассадить n вновь прибывших гостей среди m гостей, уже сидящих за круглым столом? Очевидно, что между m сидящими за круглым столом гостями имеется m промежутков, в которые можно рассаживать вновь прибывших.

Таким образом, это можно сделать

$$V(m, n) = C(m + n - 1, n) = \frac{(m + n - 1)!}{n!(m - 1)!}$$ способами.

Подстановки

В этом разделе рассматриваются подстановки и перестановки, которые на самом деле являются равнообъемными понятиями. Для вычисления количества перестановок установлена очень простая формула: $P(n) = n!$. Применяя эту формулу при решении практических задач, не следует забывать, что факториал — это очень быстро растущая функция, в частности, факториал растет быстрее экспоненты. Действительно, используя известную из математического анализа *формулу Стирлинга*

$$n! \approx \sqrt{2\pi n} n^n e^{-n}$$

$$\sqrt{2\pi n} n^n e^{-n} < n! < \sqrt{2\pi n} n^n e^{-n+1/(12n)}$$

$$\lim_{n \to +\infty} \frac{n!}{2^n} = +\infty.$$

Группа подстановок

Взаимнооднозначная функция $f\colon X \to X$ называется *подстановкой* на *X*.

Замечание

Если множество *X* конечно $(|X| = n)$, то, не ограничивая общности, можно считать, что $X = 1..n$. В этом случае подстановку $f\colon 1..n \to 1..n$ удобно задавать таблицей из двух строк. В первой строке — значения аргументов, во второй — соответствующие значения функции.

Пример

$$f = \begin{vmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 2 & 1 & 4 & 3 \end{vmatrix} \qquad g = \begin{vmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 1 & 2 & 3 & 5 \end{vmatrix}$$

Произведением подстановок *f* и *g* называется их суперпозиция *fog*.

Пример

$$fg = \begin{vmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 1 & 4 & 3 & 2 \end{vmatrix}$$

Тождественная подстановка — это подстановка *e*, такая что $e(x) = x$.

Пример

$$e = \begin{vmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{vmatrix}$$

Обратная подстановка — это обратная функция, которая всегда существует, поскольку подстановка является биекцией.

Замечание

Таблицу обратной подстановки можно получить, если просто поменять местами строки таблицы исходной подстановки.

Пример

$$f = \begin{vmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 4 & 2 & 1 & 5 \end{vmatrix} \qquad f^{-1} = \begin{vmatrix} 3 & 4 & 2 & 1 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{vmatrix} = \begin{vmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 3 & 1 & 2 & 5 \end{vmatrix}$$

Таким образом, множество подстановок образует группу относительно операции суперпозиции. Эта группа называется *симметрической* степени *n*.

Графическое представление подстановок

Подстановки удобно представлять в графической форме, проводя стрелки от каждого элемента *x* к элементу *f(x)*.

Пример
Графическое представление подстановки

$$f = \begin{vmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 1 & 4 & 5 \end{vmatrix}$$

представлено на рис. 1.

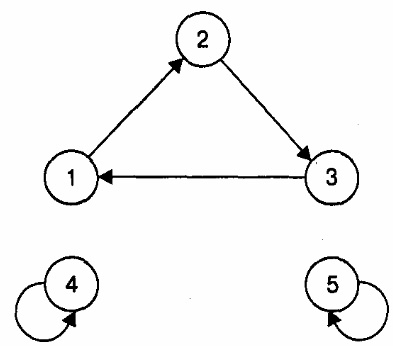

Рис. 1. Графическое представление подстановки

Циклы
Цикл — это последовательность элементов $x_0,\ldots,x_k$, такая что

$$f(x_i) = \begin{cases} x_{i+1}, & 0 \leqslant i < k, \\ x_0, & i = k. \end{cases}$$

Цикл длины 2 называется *транспозицией*.

Замечание
Из графического представления подстановки наглядно видно происхождение термина «цикл».

Подстановки и перестановки

В таблице подстановки нижняя строка (значения функции) является перестановкой элементов верхней строки (значения аргумента). Если принять соглашение, что элементы верхней строки (аргументы) всегда располагаются в определенном порядке (например, по возрастанию), то верхнюю строку можно не указывать — подстановка определяется одной нижней строкой. Таким образом, подстановки взаимно однозначно соответствуют перестановкам.
Перестановку (и соответствующую ей подстановку) элементов $1,\ldots,n$ будем обозначать $<a_i,\ldots,a_n>$, где все a_i — различные числа из диапазона $1..n$.

Инверсии

Если в перестановке $f=<a_i,\ldots,a_n>$ для элементов a_i и a_j имеет место неравенство $(a_i>a_j$, при $i < j$, то пара (a_i, a_j) называется *инверсией*. Обозначим $I(f)$ — *число инверсий* в перестановке f.

Теорема

Произвольную подстановку f можно представить в виде суперпозиции $I(f)$ транспозиций соседних элементов.

Доказательство

Пусть $f=\langle a_i,\dots, 1,\dots, a_n\rangle$ Переставим 1 на первое место, меняя ее местами с соседними слева элементами. Обозначим последовательность этих транспозиций через t_i. При этом все инверсии (и только они), в которых участвовала 1, пропадут. Затем переставим 2 на второе место и т. д. Таким образом, $f \circ t_1 \circ \dots \circ t_n = e$ и по свойству группы $f = t_n^{-1} \circ \dots \circ t_1^{-1}$, причем $|t_1| + |t_2| + \dots + |t_n| = I(f)$.

Следствие

Всякая сортировка может быть выполнена перестановкой соседних элементов.

ОТСТУПЛЕНИЕ

Доказанная теорема утверждает, что произвольную перестановку можно представить в виде композиции определенного числа транспозиций, но не утверждает, что такое представление является эффективным. Метод сортировки, основанный на предшествующей теореме, известен как *«метод пузырька»*. Заметим, что при перемещении элемента на свое место транспозициями соседних элементов все элементы остаются на своих местах, кроме перемещаемого элемента и того элемента, который стоит на целевом месте, а эти элементы меняются местами. Таким образом, метод пузырька может быть выражен в форме нижерасположенного алгоритма. Этот алгоритм прост, но является далеко не самым эффективным алгоритмом сортировки.

Вход: массив A : **array** $[1..n]$ **of** B, где значения элементов массива расположены в произвольном порядке и для значений типа B задано отношение $<$.

Выход: массив A : **array** $[1..n]$ **of** B, в котором значения расположены в порядке возрастания.

```
for i from 1 to n − 1 do
  m := i { индекс кандидата в минимальные элементы }
  for j from i + 1 to n do
    if A[j] < A[m] then
      m := j { новый кандидат в минимальные }
    end if
  end for
  A[i] ↔ A[m] { ставим минимальный элемент на место }
end for
```

Генерация перестановок

На множестве перестановок естественным образом можно определить упорядоченность на основе упорядоченности элементов. А именно, говорят, что перестановка $(a_i,\dots, a_n)$ *лексикографически* предшествует перестановке $\langle b_1, \dots, b_n\rangle$ если $\exists k \leqslant n \; a_k < b_k \,\&\, \forall i < k \; a_i = b_i$

Аналогично, говорят, что перестановка *(a_i,...,a_n) антилексикографически* предшествует перестановке $\langle b_1, \ldots, b_n \rangle$,

если $\exists k \leqslant n\ a_k > b_k \,\&\, \forall i > k\ a_i = b_i$.

Следующий алгоритм генерирует все перестановки элементов *1,..., n* в антилексикографическом порядке. Массив *P : array [l..n] of 1..n* является глобальным и предназначен для хранения перестановок.

Вход: n — количество элементов
Выход: последовательность перестановок элементов $1, \ldots, n$ в антилексикографическом порядке.

```
for i from 1 to n do
  P[i] := i { инициализация }
end for
Antilex(n) { вызов рекурсивной процедуры Antilex }
```

Основная работа по генерации перестановок выполняется рекурсивной процедурой Antilex.

Вход: m — параметр процедуры — количество первых элементов массива P, для которых генерируются перестановки.
Выход: последовательность перестановок $1, \ldots, m$ в антилексикографическом порядке.

```
if m = 1 then
  yield P { очередная перестановка }
else
  for i from 1 to m do
    Antilex(m − 1) { рекурсивный вызов }
    if i < m then
   P[i] ↔ P[m] { следующий элемент }
   Reverse(m − 1) { изменение порядка элементов }
  end if
 end for
end if
```

Вспомогательная процедура Reverse переставляет элементы заданного отрезка массива P в обратном порядке

Вход: k — номер элемента, задающий отрезок массива P, подлежащий перестановке в обратном порядке.
Выход: первые k элементов массива P переставлены в обратном порядке

```
j := 1 { нижняя граница обращаемого диапазона }
while j < k do
  P[j] ↔ P[k]
  j := j + 1
  k := k − 1
end while
```

Обоснование

Заметим следующее. Искомую последовательность перестановок *n* элементов можно получить из последовательности перестановок *n - 1* элемента следующим образом. Нужно выписать n блоков по $(n\text{-}1)!$ перестановок в каждом, соответствующих последовательности перестановок $n\text{-}1$ элемента в антилексикографическом порядке. Затем ко всем перестановкам в первом блоке нужно приписать справа n, во втором — $n\text{-}1$ и т. д. в убывающем порядке. Затем в каждом из блоков (кроме первого), к перестановкам которого справа приписан элемент i, нужно в перестановках блока заменить все вхождения элемента i на элемент n. В полученной последовательности все перестановки различны, и их $n(n\text{-}1)!=n!$, то есть перечислены все перестановки. При этом антилексикографический порядок соблюден: для последовательностей внутри одного блока, потому что этот порядок был соблюден в исходной последовательности, а для последовательностей на границах двух блоков, потому что происходит уменьшение самого правого элемента. Обратимся к процедуре Antilex — легко видеть, что в ней реализовано указанное построение. В основном цикле сначала строится очередной блок — последовательность перестановок первых $m\text{-}1$ элементов массива P (при этом элементы $P[m],..., P[n]$ остаются неизменными). Затем элемент $P[m]$ меняется местами с очередным элементом $P[i]$. Вызов вспомогательной процедуры Reverse необходим, поскольку последняя перестановка в блоке является обращением первой, а для генерации следующего блока на очередном шаге цикла нужно восстановить исходный порядок.

Пример

Последовательность перестановок в антилексикографическом порядке n=3:

$$(1,2,3),\ (2,1,3),\ (1,3,2),\ (3,1,2),\ (2,3,1),\ (3,2,1).$$

Биномиальные коэффициенты

Число сочетаний $C(m,\ n)$ — это число различных n-элементных подмножеств m-элементного множества. Числа $C(m,n)$ встречаются в формулах решения многих комбинаторных задач. Действительно, рассмотрим следующую типовую схему рассуждений при решении комбинаторной задачи. Пусть нужно определить число подмножеств m-элементного множества, удовлетворяющих некоторому условию. Разобьем задачу на подзадачи: рассмотрим отдельно 1-элементные подмножества, 2-элементые и т. д., а затем сложим полученные результаты.

Элементарные тождества

Основная формула для числа сочетаний

$$C(m,n) = \frac{m!}{n!(m-n)!}$$

позволяет получить следующие простые тождества

Теорема

1. $C(m,n) = C(m,m-n)$,
2. $C(m,n) = C(m-1,n) + C(m-1,n-1)$,
3. $C(n,i)C(i,m) = C(n,m)C(n-m,i-m)$.

Доказательство

1. $$C(m,m-n) = \frac{m!}{(m-n)!\,(m-(m-n))!} = \frac{m!}{(m-n)!\,n!} = $$
$$= C(m,n).$$

2. $$C(m-1,n) + C(m-1,n-1) =$$
$$= \frac{(m-1)!}{n!\,(m-n-1)!} + \frac{(m-1)!}{(n-1)!\,(m-1-(n-1))!} =$$
$$= \frac{(m-1)!}{n(n-1)!\,(m-n-1)!} + \frac{(m-1)!}{(n-1)!\,(m-n)(m-n-1)!} =$$
$$= \frac{(m-n)(m-1)! + n(m-1)!}{n(n-1)!\,(m-n)(m-n-1)!} =$$
$$= \frac{(m-n+n)(m-1)!}{n!\,(m-n)!} = \frac{m!}{n!\,(m-n)!} = C(m,n).$$

3. $$C(n,i)C(i,m) = \frac{n!}{i!(n-i)!} \cdot \frac{i!}{m!(i-m)!} = \frac{n!}{m!(i-m)!(n-i)!} =$$
$$= \frac{n!(n-m)!}{m!(i-m)!(n-i)!(n-m)!} =$$
$$= \frac{n!}{m!(n-m)!} \cdot \frac{(n-m)!}{(i-m)!(n-i)!} =$$
$$= C(n,m)C(n-m,i-m).$$

Бином Ньютона

Числа сочетаний *C(m,n)* называются также биномиальными коэффициентами. Смысл этого названия устанавливается следующей теоремой, известной также как формула бинома Ньютона.

Теорема

$$(x+y)^m = \sum_{n=0}^{m} C(m,n)x^n y^{m-n}.$$

Доказательство

По индукции. База, m=1:

$$(x+y)^1 = x+y = 1x^1y^0 + 1x^0y^1 = C(1,0)x^1y^0 + C(1,1)x^0y^1 = \sum_{n=0}^{1} C(1,n)x^n y^{1-n}.$$

Индукционный переход:

$$
\begin{aligned}
(x+y)^m &= (x+y)(x+y)^{m-1} = (x+y)\sum_{n=0}^{m-1} C(m-1,n)x^n y^{m-n-1} \\
&= \sum_{n=0}^{m-1} xC(m-1,n)x^n y^{m-n-1} + \sum_{n=0}^{m-1} yC(m-1,n)x^n y^{m-n-1} \\
&= \sum_{n=0}^{m-1} C(m-1,n)x^{n+1} y^{m-n-1} + \sum_{n=0}^{m-1} C(m-1,n)x^n y^{m-n} \\
&= \sum_{n=0}^{m-1} \left(C(m-1,n-1) + C(m-1,n)\right) x^n y^{m-n} + C(m-1,m-1)x^m y^0 \\
&= \sum_{n=0}^{m-1} C(m,n)x^n y^{m-n} + C(m,m)x^m y^{m-m} = \sum_{n=0}^{m} C(m,n)x^n y^{m-n}. \qquad \square
\end{aligned}
$$

Следствие

$$\sum_{n=0}^{m} C(m,n) = 2^m.$$

Доказательство

$$2^m = (1+1)^m = \sum_{n=0}^{n} C(m,n)1^n 1^{m-n} = \sum_{n=0}^{m} C(m,n).$$

Следствие

$$\sum_{n=0}^{m} (-1)^n C(m,n) = 0.$$

Доказательство

$$0 = (-1+1)^m = \sum_{n=0}^{m} C(m,n)(-1)^n 1^{m-n} = \sum_{n=0}^{m} (-1)^n C(m,n).$$

Свойства биномиальных коэффициентов

Биномиальные коэффициенты обладают целым рядом замечательных свойств

Теорема

1. $\sum\limits_{n=0}^{m} nC(m,n) = m2^{m-1}$,

2. $C(m+n,k) = \sum\limits_{i=0}^{k} C(m,i)C(n,k-i)$.

Доказательство

1. Рассмотрим следующую последовательность из чисел *1,...,т*. Сначала все подмножества длины 0, потом все подмножества длины 1 и т. д. Имеется *С(т,п)* подмножеств мощности *п,* и каждое из них имеет длину *п,* таким образом, всего в этой последовательности $\sum_{n=0}^{n} nC(m,n)$ чисел. С другой стороны, каждое число *х* входит в эту последовательность $2^{|\{1,\ldots,m\}\setminus\{x\}|} = 2^{m-1}$ раз, а всего чисел *т*.
2. *С(п + т,к)* — это число способов выбрать *к* предметов из *т* + *п* предметов. Предметы можно выбирать в два приема: сначала выбрать *i* предметов из первых *т* предметов, а затем выбрать недостающие *к—i* предметов из оставшихся *п* предметов. Отсюда общее число способов выбрать *к* предметов составляет $\sum_{i=0}^{k} C(m,i)C(n,k-i)$.

Замечание

Последнее свойство известно как *тождество Коши*.

Треугольник Паскаля

Из формулы $C(m,n) = C(m-1,n) + C(m-1,n-1)$, вытекает эффективный способ реккурентного вычисления значений биномиальных коэффициентов, который можно представить в графической форме, известной как *треугольник Паскаля* (Рис. 2.).

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

Рис. 2. Треугольник Паскаля.

В этом равнобедренном треугольнике каждое число (кроме единиц на боковых сторонах) является суммой двух чисел, стоящих над ним. Число сочетаний *С(т,п)* находится в *(т* + 1)-м ряду на *(п* + 1)-м месте.

Генерация подмножеств

Элементы множества *{!,...,т}* упорядочены. Поэтому каждое *п*-элементное подмножество также можно рассматривать как упорядоченную последовательность. На множестве таких последовательностей естественным образом определяется лексикографический порядок. Следующий простой алгоритм генерирует все n-элементные подмножества m-элементного множества в лексикографическом порядке. Ниже рассмотрим алгоритм генерации n-элементных подмножеств m-элементного множества

```
Вход: n — мощность подмножества, m — мощность множества, m ⩾ n > 0.
Выход: последовательность всех n-элементных подмножеств m-элементного множества
  в лексикографическом порядке.
  for i from 1 to m do
    A[i] := i { инициализация исходного множества }
  end for
  if m = n then
    yield A[1..n] { единственное подмножество }
    exit
  end if
  p := n { p — номер первого изменяемого элемента }
  while p ⩾ 1 do
    yield A[1..n] { очередное подмножество в первых n элементах массива A }
    if A[n] = m then
      p := p − 1 { нельзя увеличить последний элемент }
    else
      p := n { можно увеличить последний элемент }
    end if
    if p ⩾ 1 then
      for i from n downto p do
        A[i] := A[p] + i − p + 1 { увеличение элементов }
      end for
    end if
  end while
```

Обоснование

Заметим, что в искомой последовательности n-элементных подмножеств (каждое из которых является возрастающей последовательностью n чисел из диапазона $l..m$) вслед за последовательностью $(a_i,\dots,a_n)$ следует последовательность $(bi,\dots,b_n) = (a_1,\dots, a_{p-1}, a_p+1, a_p+2,\dots, a_p+n-p+1)$, где p — максимальный индекс, для которого $b_n = a_p + n - p + 1 <= m$. Другими словами, следующая последовательность получается из предыдущей заменой некоторого количества элементов в хвосте последовательности на идущие подряд целые числа, но так, чтобы последний элемент не превосходил m, а первый изменяемый элемент был на 1 больше, чем соответствующий элемент в предыдущей последовательности. Таким образом, индекс p, начиная с которого следует изменить «хвост последовательности», определяется по значению элемента $A[n]$. Если $A[n] < m$, то следует изменять только $A[n]$, и при этом $p: = n$. Если же уже $A[n] = m$, то нужно уменьшать индекс $p: =p - 1$, увеличивая длину изменяемого хвоста.

Пример

Последовательность n-элементных подмножеств m-элементного множества в лексикографическом порядке для $n = 3$ и $m = 4$: (1,2,3), (1,2,4), (1,3,4), (2,3,4).

Разбиения

Разбиения не были рассмотрены среди типовых комбинаторных конфигураций, потому что получить для них явную формулу не так просто, как для остальных. В этом разделе исследуются основные свойства разбиений, а окончательные формулы ,будут приведены далее.

Определения

Пусть $\mathcal{B} = \{B_1, \ldots, B_n\}$ есть разбиение множества X из m элементов на n подмножеств

$$B_i \subset X, \qquad \bigcup_{i=1}^{n} B_i = X, \qquad B_i \neq \varnothing, \qquad B_i \cap B_j = \varnothing \quad \text{при } i \neq j.$$

Подмножества B_i называются *блоками* разбиения.

Между разбиениями и отношениями эквивалентности существует взаимнооднозначное соответствие (см. подраздел 1.7.2). Если E_1 и E_2. — два разбиения X, то говорят, что разбиение E_1 есть *измельчение* разбиения E_2, если каждый блок E_2 есть объединение блоков E_1. Измельчение является частичным порядком на множестве разбиений.

Числа Стирлинга второго рода

Число разбиений m-элементного множества на n блоков называется *числом Стирлинга второго рода* и обозначается $S(m,n)$. По определению положим

$$\begin{aligned} S(m,0) &:= 0 \quad \text{при } m > 0, \\ S(m,m) &:= 1, \\ S(0,0) &:= 1, \\ S(m,n) &:= 0 \quad \text{при } n > m. \end{aligned}$$

Теорема

$$S(m,n) = S(m-1,n-1) + nS(m-1,n).$$

Доказательство

Пусть B – множество всех разбиений множества $\{1,\ldots,m\}$ на n блоков. Положим

$$\mathcal{B}_1 := \{X \in \mathcal{B} \mid \exists B \in X \ B = \{m\}\}, \ \mathcal{B}_2 := \{X \in \mathcal{B} \mid \neg\exists B \in X \ B = \{m\}\}$$

то есть в B_1 входят разбиения, в которых элемент m образует отдельный блок, а в B_2 – все остальные разбиения. Заметим, что

$$\mathcal{B}_2 = \{X \in \mathcal{B} \mid m \in X \Longrightarrow |X| > 1\}$$

Тогда

$$\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2, \ \mathcal{B}_1 \cap \mathcal{B}_2 = \varnothing.$$

Имеем

$$|\mathcal{B}_1| = S(m-1, n-1),\ |\mathcal{B}_2| = n\, S(m-1, n)$$

Так как все разбиения B_2 получаются следующим образом: берем все разбиения множества *{1,…,m-1}* на n блоков и в каждый блок по очереди помещаем элемент m. Следовательно

$$S(m,n) = |\mathcal{B}| = |\mathcal{B}_1| + |\mathcal{B}_2| = S(m-1, n-1) + nS(m-1, n).$$

Теорема

$$S(m,n) = \sum_{i=n-1}^{m-1} C(m-1, i) S(i, n-1).$$

Доказательство

Пусть B – множество всех разбиений множества *{1,…,m}* на n блоков. Рассмотрим семейство

$$\overline{B} := \left\{ B \subset 2^{\{1,\ldots,m\}} \mid m \in B \right\}$$

Тогда

$$\mathcal{B} = \bigcup_{B \in \overline{B}} \mathcal{B}_B,$$

где $\mathcal{B}_B := \{ X \mid X \in \mathcal{B} \,\&\, B \in X \}$, причем $\mathcal{B}_{B'} \cap \mathcal{B}_{B>>} = \varnothing$, если $B' \neq B >>$. Пусть $B \in \overline{B}$ и $b := |B|$. Тогда $|\mathcal{B}_B| = S(m-b, n-1)$

Заметим, что $\left| \left\{ B \in \overline{B} \mid |B| = b \right\} \right| = C(m-1, b-1)$

Имеем:

$$\begin{aligned} S(m,n) = |\mathcal{B}| &= \sum_{b=1}^{m-(n-1)} \left| \bigcup_{B \in \overline{B} \,\&\, |B|=b} \mathcal{B}_B \right| = \\ &= \sum_{b=1}^{m-(n-1)} C(m-1, b-1) S(m-b, n-1) = \\ &= \sum_{i=m-1}^{n-1} C(m-1, m-i-1) S(i, n-1) = \\ &= \sum_{i=n-1}^{m-1} C(m-1, i) S(i, n-1), \end{aligned}$$

где *i:=m-b*

Числа Стирлинга первого рода

Число сюръективных функций, то есть число размещений m предметов по n ящикам, таких что все ящики заняты, называется *числом Стирлинга первого рода* и обозначается *s(m,n)*.

Каждое разбиение множества *{!,… ,m}* соответствует ядру сюръективной функции и обратно. Таким образом, число различных ядер сюръективных функций — это число Стирлинга второго рода *S(m, n).* Всего сюръективных функций *s(m,n) — n!S(m,n),* так как число сюръективных функций с заданным ядром равно числу перестановок множества значений функции.

Число Белла

Число всех разбиений m-элементного множества называется *числом Белла* и обозначается *B(m).*

$$B(m) := \sum_{n=0}^{m} S(m,n), \quad B(0) := 1.$$

Теорема

$$B(m+1) = \sum_{i=0}^{m} C(m,i)B(i).$$

Доказательство

Пусть B — множество всех разбиений множества *1..m+1.* Рассмотрим множество подмножеств множества *1..m* + 1, содержащих элемент *m + 1*:

$$\overline{B} := \left\{B \subset 2^{\{1,\dots,m+1\}} \mid m+1 \in B\right\}.$$

Тогда

$$\mathcal{B} = \bigcup_{B \in \overline{B}} \mathcal{B}_B,$$

где $\mathcal{B}_B := \{X \in \mathcal{B} \mid B \in X\}.$

Пусть

$$B \in \overline{B} \text{ и } b = |B|.$$

Тогда

$$|\mathcal{B}_B| = B(m+1-b).$$

Заметим, что

$$|\left\{B \in \overline{B} \mid |B| = b\right\}| = C(m, b-1).$$

Следовательно

$$B(m+1) = |\mathcal{B}| = \sum_{b=1}^{m+1} C(m,b-1)B(m-b+1) =$$

$$= \sum_{i=m}^{0} C(m,m-i)B(i) = \sum_{i=0}^{m} C(m,i)B(i),$$

где $i = m - b + 1.$

Принцип включения и исключения

Приведенные выше алгоритмы дают способы вычисления комбинаторных чисел для некоторых распространенных комбинаторных конфигураций. Практические задачи не всегда прямо сводятся к известным комбинаторным конфигурациям. В этом случае используются различные методы сведения одних комбинаторных конфигураций к другим. В следующих разделах рассматриваются три наиболее часто используемых метода. Мы начинаем с самого простого и прямолинейного, но имеющего ограниченную сферу применения принципа включения и исключения.

Объединение конфигураций

Часто комбинаторная конфигурация является объединением других, число комбинаций в которых вычислить проще. В таком случае требуется уметь вычислять число комбинаций в объединении. В простых случаях формулы для вычисления очевидны:

$$|A \cup B| = |A| + |B| - |A \cap B|,$$

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C| + |A \cap B \cap C|.$$

Пример

Сколько существует натуральных чисел, меньших 1000, которые не делятся ни на 3, ни на 5, ни на 7? Всего чисел, меньших тысячи, 999. Из них:

- 999 : 3 = 333 делятся на 3,
- 999 : 5 = 199 делятся на 5,
- 999 : 7 = 142 делятся на 7,
- 999 : (3 * 5) = 66 делятся на 3 и на 5,
- 999 : (3 * 7) = 47 делятся на 3 и на 7,
- 999 : (5 * 7) = 28 делятся на 5 и на 7,
- 999 : (3 * 5 * 7) = 9 делятся на 3, на 5 и на 7.

Имеем: 999 - (333 + 199 + 142 - 66 - 47 - 28 + 9) = 457.

Принцип включения и исключения

Следующая формула, известная как *принцип включения и исключения*, позволяет вычислить мощность объединения множеств, если известны их мощности и мощности всех пересечений.

Теорема

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{i=1}^{n} |A_i| - \sum_{1 \leqslant i < j \leqslant n} |A_i \cap A_j| + \sum_{1 \leqslant i < j < k \leqslant n} |A_i \cap A_j \cap A_k| - \ldots$$
$$+ (-1)^{n-1} |A_1 \cap \cdots \cap A_n|.$$

Доказательство

Индукция по n. Для n=2, 3 теорема проверена уже выше. Пусть

$$\left|\bigcup_{i=1}^{n-1} A_i\right| = \sum_{i=1}^{n-1} |A_i| - \sum_{1 \leqslant i < j \leqslant n-1} |A_i \cap A_j| + \cdots + (-1)^{n-2} |A_i \cap \cdots \cap A_{n-1}|.$$

Заметим, что

$$\left(\bigcup_{i=1}^{n-1} A_i\right) \cap A_n = \bigcup_{i=1}^{n-1} (A_i \cap A_n),$$

и по индукционному предположению

$$|\bigcup_{i=1}^{n-1} (A_i \cap A_n)| = \sum_{i=1}^{n-1} |A_i \cap A_n| - \sum_{1 \leqslant i < j \leqslant n-1} |A_i \cap A_j \cap A_n| + \ldots$$
$$+ (-1)^{n-2} |A_1 \cap \cdots \cap A_{n-1} \cap A_n|.$$

Тогда

$$\left|\bigcup_{i=1}^{n} A_i\right| = \left|(\bigcup_{i=1}^{n-1} A_i) \cup A_n\right| = \left|\bigcup_{i=1}^{n-1} A_i\right| + |A_n| - \left|(\bigcup_{i=1}^{n-1} A_i) \cap A_n\right| =$$
$$= \left(\sum_{i=1}^{n-1} |A_i| - \sum_{1 \leqslant i < j \leqslant n-1} |A_i \cap A_j| + \cdots + (-1)^{n-2} |A_1 \cap \cdots \cap A_{n-1}|\right) +$$
$$+ |A_n| - \left(\sum_{i=1}^{n-1} |A_i \cap A_n| - \sum_{1 \leqslant i < j \leqslant n-1} |A_i \cap A_j \cap A_n| + \ldots\right.$$
$$\left. + (-1)^{n-2} |A_1 \cap \cdots \cap A_{n-1} \cap A_n|\right) =$$
$$= \left(\sum_{i=1}^{n-1} |A_i| + |A_n|\right) - \left(\sum_{1 \leqslant i < j \leqslant n-1} |A_i \cap A_j| + \sum_{i=1}^{n-1} |A_i \cap A_n|\right) + \ldots$$
$$- (-1)^{n-2} |A_1 \cap \cdots \cap A_{n-1} \cap A_n| =$$
$$= \sum_{i=1}^{n} |A_i| - \sum_{1 \leqslant i < j \leqslant n} |A_i \cap A_j| + \cdots + (-1)^{n-1} |A_1 \cap \cdots \cap A_n|.$$

Число булевых функций, существенно зависящих от всех своих переменных

Рассмотрим применение принципа включения и исключения на примере следующей задачи. Пусть $p_n := |P_n| = 2^{2^n}$ — число всех булевых функций n переменных, а p_n — число булевых функций, существенно зависящих от всех n переменных. Пусть P_n^i — множество булевых функций, у которых переменная Xi фиктивная (кроме x_t могут быть и другие фиктивные переменные).

Имеем:

$$\widetilde{p}_n = |P_n \setminus (P_n^1 \cup \cdots \cup P_n^n)| = \left|P_n \setminus \bigcup_{n=1}^{n} P_n^i\right|.$$

С другой стороны $|P_n^i| = 2^{2^{n-1}}$

Более того
$$|P_n^{i_1}\cap\cdots\cap P_n^{i_k}| = 2^{2^{n-k}}.$$
Следовательно
$$\begin{aligned}\widetilde{p}_n &= 2^{2^n} - \left(\sum_{i=1}^{n}|P_n^i| - \sum_{1\leqslant i<j\leqslant n}|P_n^i\cap P_n^j| + \cdots + (-1)^{n-1}|P_n^1\cap\cdots\cap P_n^n|\right) = \\ &= 2^{2^n} - \left(C(n,1)2^{2^{n-1}} - C(n,2)2^{2^{n-2}} + \cdots + (-1)^{n-1}C(n,n)2\right) = \\ &= \sum_{i=0}^{n}(-1)^i C(n,i)2^{2^{n-i}}.\end{aligned}$$

Формулы обращения

Очень полезную, но специфическую группу приемов образуют различные способы преобразования уже полученных комбинаторных выражений. В этом разделе рассматривается один частный, но важный случай.

Теорема обращения

Пусть $a_{n,k}$ *и* $b_{n,k}$ — некоторые (комбинаторные) числа, зависящие от параметров n и κ, причем $0 <= \kappa <= n$. Если известно выражение чисел $a_{n,\kappa}$ через числа $b_{n,\kappa}$, то в некоторых случаях можно найти и выражение чисел $b_{n,\kappa}$ через числа $a_{n,\kappa}$, то есть решить комбинаторное уравнение.

Теорема

Пусть
$$\forall n\ \forall k \leqslant n\ a_{n,k} = \sum_{i=0}^{n}\lambda_{n,k,i}b_{n,i}$$
и пусть
$$\exists\,\mu_{n,k,i}\ \forall k\leqslant n\ \forall m\leqslant n\ \sum_{i=0}^{n}\mu_{n,k,i}\lambda_{n,i,m} = \begin{cases}1, & m=k,\\ 0, & m\neq k.\end{cases}$$
Тогда
$$\forall k\leqslant n\ b_{n,k} = \sum_{i=0}^{n}\mu_{n,k,i}a_{n,i}.$$

Доказательство
$$\begin{aligned}\sum_{i=0}^{n}\mu_{n,k,i}a_{n,i} &= \sum_{i=0}^{n}\mu_{n,k,i}\left(\sum_{m=0}^{n}\lambda_{n,i,m}b_{n,m}\right) = \\ &= \sum_{m=0}^{n}\left(\sum_{i=0}^{n}\mu_{n,k,i}\lambda_{n,i,m}\right)b_{n,m} = b_{n,k}.\end{aligned}$$

Формулы обращения для биномиальных коэффициентов

Применение теоремы обращения предполагает отыскание для заданных чисел $l_{n,k,i}$ (коэффициентов комбинаторного уравнения) соответствующих чисел, $m_{n,k,i}$ удовлетворяющих условию теоремы обращения. Особенно часто числами $l_{n,k,i}$ являются биномиальные коэффициенты.

Лемма

$$\sum_{i=0}^{n}(-1)^{i-m}C(n,i)C(i,m) = \begin{cases} 1, & m = n, \\ 0, & m < n. \end{cases}$$

Доказательство

$$\begin{aligned} \sum_{i=0}^{n}(-1)^{i-m}C(n,i)C(i,m) &= \sum_{i=0}^{n}(-1)^{i-m}C(n,m)C(n-m,i-m) = \\ &= \sum_{i=m}^{n}(-1)^{i-m}C(n,m)C(n-m,i-m) = \\ &= C(n,m)\sum_{i=m}^{n}(-1)^{i-m}C(n-m,i-m). \end{aligned}$$

Но при $m<n$ имеем:

$$\sum_{i=m}^{n}(-1)^{i-m}C(n-m,i-m) = \sum_{j=0}^{n-m}(-1)^{j}C(n-m,j) = 0,$$

где $j:=i-m$. С другой стороны, при $m=n$ имеем:

$$C(n,m)\sum_{i=m}^{n}(-1)^{i-m}C(n-m,i-m) = C(n,n)(-1)^{n-n}C(0,0) = 1.$$

Теорема

$$\textit{Если } a_{n,k} = \sum_{i=0}^{k} C(k,i)b_{n,k}, \textit{ то } b_{n,k} = \sum_{i=0}^{k}(-1)^{k-i}C(k,i)a_{n,i}.$$

Доказательство

Здесь

$$\lambda_{n,k,i} = C(k,i) \text{ и } \mu_{n,k,i} = (-1)^{k-i}C(k,i).$$

При $k<=n$, $m<=n$ имеем:

$$\sum_{i=0}^{n}\mu_{n,k,i}\lambda_{n,i,m}=\sum_{i=0}^{n}(-1)^{i-k}C(i,k)C(m,i)=$$
$$=\sum_{i=0}^{n}(-1)^{i-k}C(m,i)C(i,k)=$$
$$=\begin{cases}1, & k=n,\\ 0, & k\neq n.\end{cases}$$

Формулы для чисел Стирлинга

В качестве примера использования формул обращения рассмотрим получение явных формул для чисел Стирлинга первого и второго рода. Рассмотрим множество функций $f{:}A\to B$, где $|A|=n$ и $|B|=k$. Число всех таких функций равно κ^n. С другой стороны, число функций f, таких что $|f(A)| = i$, равно $s(i, n)$, поскольку $s(i,n)$ — это число сюръективных функций $f{:}1..n \to 1..i$. Но область значений функции (при заданном i) можно выбрать $C(k,i)$ способами. Поэтому

$$k^n=\sum_{i=0}^{k}C(k,i)s(i,n).$$

Обозначив $a_{n,k}:=k^n$ и $b_{n,i}:=s(i,n)$, , имеем, основываясь на теореме, приведенной выше

$$s(k,n)=\sum_{i=0}^{n}(-1)^{k-i}C(i,k)i^n.$$

Учитывая связь чисел Стирлинга первого и второго рода, имеем:

$$S(k,n)=\frac{1}{n!}\sum_{i=0}^{n}(-1)^{k-i}C(i,k)i^n.$$

Производящие функции

Для решения комбинаторных задач в некоторых случаях можно использовать методы математического анализа. Разнообразие применяемых здесь приемов весьма велико, и не может быть в полном объеме рассмотрено в рамках этой книги. В данном разделе рассматривается только основная идея метода производящих функций, применение которой иллюстрируется простым примером.

Основная идея

Пусть есть последовательность комбинаторных чисел a_4 и последовательность функций $j_i(x)$. Рассмотрим формальный ряд:

$$\mathfrak{F}(x):=\sum_{i}a_i\varphi_i(x).$$

$\mathcal{F}(x)$ называется *производящей функцией* (для заданной последовательности комбинаторных чисел a_i относительно заданной последовательности функций $j_i(x)$).

Обычно используют $\varphi_i(x) := x^i$ или $\varphi_i(x) := x^i/i!$. .

Пример

Из формулы бинома Ньютона при $y := 1$ имеем:

$$(1+x)^n = \sum_{i=0}^{n} C(n,i)x^i.$$

Таким образом, $(1 + x)^n$ является производящей функцией для биномиальных коэффициентов.

Метод неопределенных коэффициентов

Из математического анализа известно, что если

$$\mathcal{F}(x) := \sum_i a_i\varphi_i(x) \text{ и } \mathcal{F}(x) := \sum_i b_i\varphi_i(x),$$

то $\forall i\ a_i = b_i$ (для рассматриваемых здесь систем функций j_i).

В качестве примера применения производящих функций рассмотрим доказательство следующего тождества.

Теорема

$$C(2n,n) = \sum_{k=0}^{n} C(n,k)^2.$$

Доказательство

Имеем:

$$(1+x)^{2n} = (1+x)^n(1+x)^n.$$

Следовательно,

$$\sum_{i=0}^{2n} C(2n,i)x^i = \sum_{i=0}^{n} C(n,i)x^i \cdot \sum_{i=0}^{n} C(n,i)x^i.$$

Приравняем коэффициент при x^n:

$$C(2n,n) = \sum_{k=0}^{n} C(n,k)C(n,n-k) = \sum_{k=0}^{n} C(n,k)^2.$$