векторами взаимно однозначно, то подмножеств столько же, сколько n-мерных двоичных векторов, т.е. 2^n.

О кодировании

Рассмотренная задача показывает, какую роль играет удачное кодирование объектов; часто оно является основным звеном решения вопроса. Наоборот, неудачное кодирование обычно заводит в тупик. Вернемся к нашей простенькой задаче о распределениях (произвольных) трех различных конфет между тремя лицами. Выше мы кодировали распределение кортежами (a_1, a_2, a_3), где a_k – лицо, получившее k-ю конфету. Нельзя ли кодировать не «по конфетам», а «по лицам», обозначая через a_k то, что получает k-е лицо (k = 1, 2, 3)?

Каждая компонента при этом будет подмножеством (множества из 3 конфет) и может сама по себе принимать $2^3 = 8$ значений. Как уже говорилось, компоненты кортежей могут иметь любую природу, им не возбраняется быть и подмножествами. Хуже другое: при таком кодировании нельзя применить принцип умножения, так как не выполняется его основное условие. Например, если $a_1 = \varnothing$ (пустое множество), т.е. первому не досталось ничего, то для a_2 остаются те же 8 вариантов, если же первый получает все, то для a_2 возможен лишь один вариант ($\varnothing$). Так что кодирование по лицам является неудачным; это различие между двумя способами кодирования исчезает при «справедливых» распределениях (каждому по конфете), когда лица и конфеты входят в задачу равноправно.

Разумеется, всегда надо следить за взаимно однозначным соответствием между объектами и кодами – каждому объекту (из нашего множества) должен отвечать ровно один код, и обратно. Только в этом случае подсчет числа объектов можно заменить подсчетом числа кодов.

Размещения

Далее речь будет часто идти о кортежах, все компоненты которых принимают значения из одного и того же непустого конечного множества. Иначе говоря, мы будем иметь дело со словами в некотором алфавите. Разумеется, в качестве «букв» алфавита могут выступать и числа – например, 1, 2, …, n. Так как элементы конечного множества всегда можно занумеровать, то удобная алфавитно-буквенная терминология не меняет сути дела.

Как уже говорилось, число m-буквенных слов в n-буквенном алфавите равно n^m. Иногда это число называют числом размещений из n по m с повторениями. Обычно же термин «размещение» связывается с требованием, чтобы все буквы слова были различны (не повторялись).

Задача 4. *Сколько существует в n-буквенном алфавите m-буквенных слов, состоящих из различных букв?*

Задача представляет интерес лишь при $m \le n$ (иначе ответ, очевидно, нуль). Можно представить себе, что имеется n карточек с изображением всех букв алфавита (по одной карточке на букву) и мы формируем слова, размещая друг за другом те или иные m карточек в том или ином порядке.

Задача легко решается с помощью принципа умножения, первая буква a_1 может пробегать n значений, при любой фиксированной a_1 для a_2 остается $n - 1$ значений (все буквы алфавита, кроме a_1) и т.д., вплоть до последней буквы, для которой остается $n - m + 1$ значений. Итак, искомое число равно

$$n(n-1)\ldots(n-m+1) = \frac{n!}{(n-m)!} \quad (m \le n). \qquad (3)$$

Здесь, как и всегда в математических выражениях, $k!$ (читается «ка факториал») есть сокращенная запись произведения $1 \cdot 2 \cdot \ldots \cdot k$, где k – любое натуральное число (например, $4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$). По определению полагается также $0! = 1$.

Величина (3) называется числом размещений из n элементов по m и часто обозначается через A_n^m. Если нет специальной оговорки о повторениях, то под числом размещений из n по m всегда подразумевается величина (3).

Пример: если в турнире участвуют 20 команд и дележ мест исключен, то первые 3 места (т.е. золотая, серебряная, бронзовая медали) могут распределяться

$$A_{20}^3 = 20 \cdot 19 \cdot 18 = 6840$$

способами. Повторения здесь, очевидно, невозможны, так как одна команда не может занять сразу два места.

Перестановки

Это просто частный случай размещений при $m = n$. Итак, число перестановок из n элементов есть $n!$. Например, в алфавите $\{a, b, c\}$ имеется $3! = 6$ перестановок: *abc, acb, bac, bca, cab, cba.* С ростом n величина $n!$ быстро растет. В отличие от размещений (при $m < n$), перестановки отличаются друг от друга только порядком букв, но не «буквенным составом». Мы просто переставляем те же самые n карточек в том или ином порядке, отсюда и название.

Вернемся к турниру 20 команд из предыдущего пункта. Если интересоваться распределением всех мест (а не только трех первых), то число возможных вариантов становится равным $20!$ ($> 2 \cdot 10^{18}$). Любопытно, что перебор всех этих вариантов мощным компьютером, обрабатывающим миллиард вариантов в секунду, займет больше времени, чем реальное проведение турнира – даже если проводить по одному туру в год!

В теории дискретной оптимизации хорошо известна так называемая *задача коммивояжера*. Речь в ней идет о городах $T_1, T_2, \ldots, T_n$, расстояние между которыми заданы; требуется выбрать такой порядок объезда этих городов, начиная с T_1, чтобы суммарная длина маршрута была минимальной. Поскольку число маршрутов конечно, казалось бы, тут нет проблемы – просто перебрать все варианты и выбрать наилучший, тем более если под рукой хороший компьютер. После всего сказанного ранее читатель, несомненно, догадывается, в чем тут загвоздка. Перебирать ведь придется $(n - 1)!$ маршрутов – число, хотя и конечное, но с ростом n быстро становящееся «практически бесконечным». При $n = 5$ компьютер выдаст кратчайший маршрут мгновенно, при $n = 15$ провозится куда больше, чем хотелось бы, а при $n = 20$ до получения ответа можно и не дожить…

Мы говорим о гигантских, мучительно долгих вычислениях, а сами, между прочим, подсчитываем все очень легко, можно сказать, на пальцах. Нет ли здесь противоречия? Нет, конечно. Мы ведь только находим число вариантов, применяя при этом «сверхскоростные» (по сравнению с примитивным перебором) приемы комбинаторики. Вместо самого перебора мы лишь прикидываем, сколько времени он бы занял. Чтобы пройти