**Гашков Сергей Борисович**
*Доктор физико-математических наук,*
*профессор кафедры дискретной математики*
*механико-математического факультета МГУ*
*им. М.В. Ломоносова*

# Упаковка рюкзака и размен монет

Продолжим разговор о математике, встречающейся в обыденной жизни.

Всем приходилось запихивать в рюкзак (или сумку, или ящик, контейнер, грузовик и т.п.) разнообразные предметы с целью их перевозки или переноски.

В этом занятии тоже можно разглядеть интересные математические задачи.

Некоторые из них по сути геометрические и могут быть очень трудными. Примеры таких задач приведены в статье из прошлого номера, хотя там речь шла не о рюкзаке, а о пенале.

Удивительно, что очень трудными могут оказаться и простейшие (одномерные) варианты таких задач, которые можно сформулировать как задачи о размещении в данном отрезке данного множества меньших отрезков. В более привычной формулировке эти задачи выглядят так: дан рюкзак с данной максимальной вместимостью, например 30 кг, в него надо упаковать предметы с данными весами, например 1 кг, 3 кг и т.д.

## Варианты задачи об упаковке рюкзака

Возникают такие вопросы: как упаковать в рюкзак наибольшее возможное число предметов из данного списка, или как упаковать в него набор предметов максимального возможного суммарного веса (или стоимости, если вы собираетесь ограбить магазин)? Каждая из этих двух задач имеет два варианта: когда в рюкзак можно класть и несколько одинаковых предметов, и когда в упаковке все предметы должны быть различны (эта ситуация возникает, например, при ограблении картинной галереи, только вместо рюкзака нужен грузовик). Далее, в каждом из вариантов задачу можно сформулировать опять в двух вариантах: в одном из них надо лишь определить даже не величину максимальной загрузки рюкзака (суммарный вес или число предметов), а всего лишь ответить на вопрос, может ли эта загрузка быть больше некоего заданного числа, а в чуть более сложном надо конкретно указать и величину максимальной загрузки и сам набор составляющих ее предметов (в случае реальных трехмерных задач упаковки надо еще и указать, как конкретно будут размещаться предметы, но здесь мы рассматриваем простейший одномерный вариант, в котором эта проблема не возникает).

Конечно, все эти задачи можно сформулировать математически как некоторые задачи о решении уравнений и неравенств, иногда с дополнительными условиями.

Например, задачу о возможности упаковать не менее $k$ предметов с весами $w_1, \ldots, w_n$ в рюкзак вместимости $K$ можно сформулировать так: имеет ли решение система неравенств

$$\begin{cases} x_1 w_1 + \ldots + x_n w_n \leq K, \\ x_1 + \ldots + x_n \geq k. \end{cases}$$

Следует уточнить, что понимается под неизвестными $x_i$. Они могут принимать любые целые неотрицательные значения (нуль означает, что предметов с весом $w_i$ в упаковке нет), или только значения 0 и 1 (если рассматриваем вариант с запретом брать в упаковку одинаковые предметы). В первом варианте эту задачу для краткости называют целочисленным рюкзаком. Во втором случае говорят, что речь идет о булевом рюкзаке (потому что неизвестные принимают булевы значения 0 и 1, так для краткости их называют в честь английского математика Джорджа Буля). Оба рассмотренных варианта относятся к типу задач распознавания, потому что в качестве ответа ожидается лишь да или нет, в зависимости от того, имеет ли система решение. Само решение в задачах типа распознавание предъявлять не надо. Его требуется находить в задачах оптимизации, например, в оптимизационном варианте рассмотренной задачи нужно найти решение первого неравенства, которое максимизирует сумму переменных (т.е. общее число предметов).

Можно рассмотреть вариант задачи об упаковке, в котором надо максимизировать сумму весов, т.е.

$$x_1 w_1 + \ldots + x_n w_n,$$

при условии выполнения неравенства

$$x_1 w_1 + \ldots + x_n w_n \leq K.$$

Опять этот вариант можно рассматривать в булевом и не булевом случае.

Можно рассмотреть и более общую задачу, в которой надо максимизировать не суммарный вес, а суммарную стоимость. Тогда в формальной записи условия, приведенной выше, в нижнем неравенстве надо перед неизвестными поставить заданные положительные коэффициенты.

Нужно еще уточнить, какие коэффициенты используются в рассмотренных системах неравенств. Очевидно, что практический смысл имеют только рациональные коэффициенты, поэтому их можно просто считать натуральными числами (к этому случаю задача сводится путем умножения на общий знаменатель всех дробей).

Кажется, что формулировка задач об упаковке в виде систем линейных неравенств (так называются неравенства, в которых неизвестные входят только в первой степени) показывает, что эти задачи являются частными случаями задач линейного программирования. Но это не совсем так, в них неизвестные принимают целые значения, поэтому они являются частными случаями задачи целочисленного линейного программирования. На первый взгляд, ограничение целыми неизвестными только облегчает задачу, позволяя решить ее простым перебором. Однако этот перебор имеет, как говорят, экспоненциальный размер.

Например, уже в частном случае задачу о булевом рюкзаке, которую можно переформулировать как задачу о разрешимости данного линейного уравнения с натуральными коэффициентами

$$x_1w_1+\ldots+x_nw_n=K$$

в числах 0,1. Эту задачу иногда называют задачей о точном булевом рюкзаке или задачей о подсумме.

## Проблема перебора

В рассматриваемой задаче лобовой перебор возможных комбинаций значений неизвестных с целью поиска хотя бы одного решения этого уравнения (т.е. выяснения, возможно ли точное заполнение рюкзака) имеет размер $2^n$. Это конечно верхняя граница размера перебора, ее можно улучшить, но как полностью устранить перебор в этой и подобных задачах сейчас не знает никто. Конечно, при небольших значениях $n$ и небольших весах $w_i$ этот перебор легко выполняется. Но уже если $n = 10$ его размер превышает тысячу и ручной перебор становится затруднительным. При $n = 30$ может вызвать трудности и у суперкомпьютера. А при $n = 100$ даже суперкомпьютер будет его выполнять миллиарды лет

Уточним, что под экспоненциальной оценкой обычно понимают оценку вида $a^n$, где $a > 1$, а оценку вида $n^a$ называют степенной, а чаще — полиномиальной. Очевидно, что при достаточно больших $n$ любая экспоненциальная оценка будет больше любой полиномиальной. Число $n$ в таких оценках является характеристикой размера задачи.

Иногда понятие размера задачи требуется уточнить. Рассмотрим, например, задачу проверки простоты данного числа $n$. Она очевидно решается путем перебора всех возможных делителей от 2 до квадратного корня из $n$.

Проверка делимости требует числа операций порядка квадрата длины двоичной записи числа $n$. Значит, общее число операций для проверки числа $n$ на простоту оценивается по порядку числом $n$, т.е. оценка времени тестирования на простоту полиномиальна. Но дело в том, что настоящим размером числа $n$ является размер его двоичной записи, т.е. приблизительно двоичный логарифм $n$. А относительно этого, уже правильного размера задачи, время работы приведенного тривиального алгоритма уже будет экспоненциальным.

Поэтому правильной оценкой размера задачи о булевом (или целочисленном) рюкзаке является сумма логарифмов весов $w_i$. Но она очевидно не меньше $n$, поэтому число $n$ можно использовать в качестве нижней оценки размера задачи. Относительно нее очевидный переборный алгоритм решения любого из рассмотренных вариантов задачи об упаковке рюкзака будет работать как минимум экспоненциальное время.

Приведем еще пример возникающей в жизни задачи, связанной с комбинаторным перебором. Эта задача о том, как объехать данные $n$ пунктов (например, городов или магазинов), вернуться назад, и затратить на это минимальное время (или минимальные деньги). Если уж мы упаковали рюкзак, то можно отправится в турпоездку в интересные города, или обойти все магазины в округе с целью заполнить рюкзак покупками.

Эта задача известна как задача коммивояжера (странствующего торговца). Для простоты предположим, что расстояние между любыми двумя пунктами равно 1 или 2 (или не расстояние, а время в пути между ними). Тогда задача сводится к нахождению замкнутого маршрута, проходящего по всем городам по одному разу, для которого сумма весов его ребер минимальна (под ребром понимаем участок пути между пунктами, а под весом — его длину, т.е. число 1 или 2.) Очевидный способ решения требует перебора всех $(n-1)! = (n-1)(n-2)\ldots 2$ замкнутых маршрутов, проходящих по разу через каждый пункт. Функция $n!$ ($n$ факториал) растет даже быстрее, чем любая экспонента $a^n$ (можно оценить ее снизу как $(n/3)^n$, а сверху как $(n/2)^n$, и даже гораздо более точно, но нам это здесь не понадобится).

Можно ли всегда найти решение этой задачи за полиномиальное время, никому не известно.

Поясним, почему в качестве размера задачи было взято число $n$. Строго говоря, размер этой задачи равен $n(n-1)/2$ (потому, что она полностью определяется указанием расстояний между всеми парами пунктов, а эти расстояния равны 1 или 2, или, если угодно, 0 и 1, так как решение задачи от такой замены не меняется), но очевидно, что полиномиальная оценка относительно $n$ остается полиномиальной и относительно $n(n-1)/2$.

Задачи, которые решаются всегда за полиномиальное время кратко называют полиномиальным, а класс всех таких задач обозначают буквой $P$. Такие задачи считаются быстро решаемыми. Конечно, задачи, решаемые за время $O(n)$ (такое обозначение используют для краткости вместо обозначения $Cn$, где $C$ — некоторая константа, не зависящая от $n$) действительно быстро решаемые, как и задачи, требующие время $O(n \log n)$ (если, конечно, константы С, спрятанные за буквой $O$, не астрономических размеров). Такие задачи называют задачами, решаемыми за линейное или почти линейное время. Задачи, решаемые за время $O(n^2)$ называют задачами квадратичной сложности, а задачи, требующие время $O(n^3)$ называют задачами кубичной сложности. Встречаются, конечно, и более сложные задачи.

Не ясно, принадлежат ли к классу $P$ рассмотренные задачи об упаковке рюкзака и задача коммивояжера. Но они заведомо относятся к классу задач, которые можно было бы решить за полиномиальное время при наличии подсказки. Например, если мы хотим узнать, можно ли полно упаковать рюкзак данными предметами, достаточно, чтобы кто-нибудь нам указал, какие предметы нужно в него класть. А уж проверить, полностью ли он заполняется этим предметами, мы сможем быстро. А если мы хотим узнать, например, существует ли маршрут между $n$ пунктами суммарной длины не более данного числа $K$ (в рассматриваемом примере интересен случай, когда $n \le K < 2n$, в остальных случаях ответ очевиден), то, если нам кто-то укажет подходящий маршрут, нам будет достаточно проверить, что он проходит по разу по каждому городу, найти его длину и сравнить с числом $K$. Все это очевидно делается за полиномиальное время относительно $n$.

По традиции указанный класс задач, к которому принадлежат задачи о рюкзаке и коммивояжере, называют $NP$. Без подсказки они очевидно решаются перебором (фактически перебором всех подсказок), но это требует экспоненциального или даже большего времени. Можно назвать задачи из класса $NP$ также переборными задачами. По определению класс $P$ содержится в $NP$ (задачи из класса $P$ быстро решаются без подсказки, т.е. без перебора). Возможно, что эти классы совпадают. Тогда кто-нибудь придумает быстрый алгоритм для решения переборных задач, и их уже не будут так называть.

Но большинство математиков склоняется к мысли, что класс $NP$ шире класса $P$. За доказательство или опровержение этой гипотезы институт Клея в США предлагает премию в миллион долларов. Но пока никто на нее не претендует (возможно, кроме некоторых фриков, утверждающих, что они доказали равенство $P = NP$).

Однако в классе NP выделен класс задач (американским математиком Стивеном Куком), которому дано даже строгое определение (которое довольно сложное, и поэтому опускается), называемых $NP$-полными. Они так называются, потому что любую задачу из класса $NP$ можно в определенном смысле (это определение также здесь опускается) полиномиально свести к любой $NP$-полной задаче (что и объясняет это название). Отсюда выводится, что $NP$-полные задачи сравнимы друг с другом по сложности, так как они полиномиально сводятся друг к другу, и являются самыми сложными среди всех задач из класса $NP$. Очевидно, что если задача является более общей, чем $NP$-полная задача, то эта более общая задач тоже $NP$-полна.

Например, все рассмотренные варианты задачи об упаковке рюкзака, являются частными случаями общей задачи целочисленного линейного программирования. В ней дается произвольная система линейных равенств и неравенств, и некоторая линейная функция, которую нужно максимизировать или минимизировать в предположении, что неизвестные принимают только целые значения. Доказано, что задача об упаковке рюкзака $NP$-полна. Поэтому общая задача целочисленного программирования тоже $NP$-полна.

А про частный случай $NP$-полной задачи вообще говоря нельзя утверждать, что он $NP$-полон. Это требует отдельного доказательства или может оказаться неверным. Например, некоторые частные случаи булевого рюкзака полиномиальны. А общая задача булева программирования (так называют частный случай целочисленного программирования, в котором переменные ограничены значениями 0 и 1) $NP$-полна, потому что ее частным случаем является булев рюкзак.

На первый взгляд кажется, что булев рюкзак — это частный случай целочисленного рюкзака, но это не так. В булевом рюкзаке более жесткие ограничения на переменные. Если найдено булево решение задачи о рюкзаке, то тем самым найдено и целочисленное решение, но не наоборот. В булевом варианте размер перебора может быть меньше, но если не найдено булево решение, это не значит, что нет целочисленного решения, а если это оптимизационные задачи, то результаты в них могут быть разными, даже если все коэффициенты совпадают. Поэтому умение решать булев вариант не означает умения решать целочисленный вариант и наоборот.

Интересно, что и даже некоторые частные случаи задачи о булевом рюкзаке (или о подсумме) трудно решаемы. Например, если в общей задаче о подсумме, в которой требуется найти булевой решение уравнения

$$W_1x_1+\ldots+W_nx_n=K,$$

взять $K = M/2$, где $M = W_1+ \ldots + W_n$, то она тоже будет трудно решаемой. Эта задача называется задачей о разбиении, потому что в ней фактически речь идет о том, как разделить предметы со стоимостями $W_i$ между двумя наследниками поровну (один берет те предметы, для которых $x_i=1$, а другой — все остальные).

Покажем, что общая задача о подсумме сводится к своему частному случаю — задаче о разбиении, поэтому задача о разбиении тоже будет трудно решаемой.

Для этого добавим к числам $W_i$ еще два числа $W_{n+1}= 3M - 2K$ и $W_{n+2}=2M$.

*Докажите, что все $n+2$ указанных числа можно разбить на две равные суммы тогда и только тогда когда из чисел $W_1, \ldots, W_n$ можно составить подсумму, равную $K$.*

Указание. Очевидно $M > K$. Если $n+2$ числа разбиты на равные подсуммы, то числа $2M$ и $3M - 2K$ находятся в разных подсуммах (так как $5M - 2K > 3M > M$). Тогда одна подсумма равна $A + 3M - 2K$, а вторая равна $(M - A) + 2M$, и из их равенства следует, что подсумма $A = K$.

Сейчас усилиями сотен математиков найдены уже тысячи $NP$-полных задач. Все рассмотренные варианты задачи упаковки в рюкзак и задачи коммивояжера тоже оказались $NP$-полными. Если для какой-нибудь из них доказать, что она не полиномиальна, то класс $NP$ будет шире класса $P$ хотя бы за счет $NP$-полных задач (но останется тогда возможность существования не $NP$-полных, но и не полиномиальных задач). Если же для какой-нибудь $NP$-полной задачи доказать ее полиномиальность, то класс $NP$ просто совпадет с классом $P$. Так как в это сейчас мало кто верит, то $NP$-полные задачи иногда называют трудно решаемыми.

## Задача о размене монет

Частным случаем задачи об упаковке рюкзака является задача о точном целочисленном рюкзаке, т.е. о том, можно ли заполнить рюкзак полностью данным набором предметов.

Другое ее название — задача о размене монет (действительно, в ней идет речь о том, можно ли данную сумму разменять монетами заданных номиналов)

Эта задача, как оказалось, тоже трудно решаемая, но доказать это довольно трудно, делается это сведением к ней задачи о булевом рюкзаке (это значит, что можно показать как быстро решить любую задачу о булевом рюкзаке, если уметь быстро решать задачу о размене монет).

На первый взгляд эта задача даже легче, чем задача проверки разрешимости в целых числах линейного уравнения с целыми коэффициентами. А эта задача легко решается за полиномиальное время, так как достаточно вычислить наибольший общий делитель коэффициентов уравнения с помощью алгоритма Евклида и проверить, будет ли он делителем числа в другой части уравнения. Например, если дано уравнение $3x + 5y + 7z = 1000$, то убедившись, что наибольший общий делитель коэффициентов равен 1, и он очевидно делит правую часть, то согласно известной теореме теории чисел это уравнение имеет решение в целых числах, и даже бесконечно много решений. Но задача о решении его в целых неотрицательных числах не так уж и проста, и указанная теорема здесь сразу на вопрос не дает ответа.

Разумеется, частные случаи общей задачи о размене монет могут легко и даже очевидно решаться, например, когда среди монет $a_i$ имеется монета с единичным номиналом.

Ясно также, что общая задача быстро сводится к частному случаю, когда наибольший общий делитель всех номиналов ($a_1$,..., $a_n$) равен единице (иначе можно все номиналы поделить на их общий делитель).

Далее рассматриваем только этот случай. Можно доказать, что найдется число $f= f(a_1,..., a_n)$ такое, что сумму в $f$ таньга нельзя разменять монетами с номиналами $a_1,...,$ $a_n$ таньга, а любую большую сумму — можно (попробуйте доказать это утверждение — это не такая уж и легкая задача).

Функция $f(a_1,...,a_n)$ называется функцией Фробениуса, в честь известного немецкого математика, поставившего проблему вычисления этой функции. Если бы был известен быстрый алгоритм решения задачи размена монет, то эта проблема наверное уже была бы решена. Неудивительно, что она оказалась очень трудной, хотя в частных случаях значение этой функции вычислить можно. Читатель сам может убедиться в трудности этой задачи, попробовав доказать следующие утверждения.

Например, если $a_1= n^2- n + 2$, $a_2=a_1+1$, ..., $a_n= a_1+n-1= n^2+1$, то $f(a_1,..., a_n)=n^3+n+1$.
(это задача с одной из австрийских олимпиад).

Известный американский математик Дж. Сильвестр еще до Фробениуса доказал, что $f(a, b) = ab - a - b$ (в 1979 г. это предлагалось доказать старшеклассникам на олимпиаде во Франции). В частности, это означает, что трешками и пятерками можно уплатить любую сумму, начиная с 8 рублей, а 7 рублей — нельзя.

Следующая задача предлагалась на IMO 1982. *Пусть $a$, $b$, $c$ попарно взаимно просты (т.е. для каждой пары из этих трех чисел утверждается их взаимная простота). Докажите, что любую сумму, начиная с $2abc - ab - ac - bc + 1$, можно уплатить монетами достоинством $ab$, $ac$ и $bc$, а сумму $2abc - ab - ac - bc$ — нельзя (т.е. $f(ab, ac, bc) = 2abc - ab - ac - bc$).*

## Булев рюкзак и криптография

В некоторых случаях задача о точном булевом рюкзаке легко решается.

Если последовательность $a_i$ быстро растущая, т.е. удовлетворяет всем неравенствам

$$a_i > a_{i-1} + \ldots + a_1,$$

то эта задача решается за полиномиальное время жадным алгоритмом.

Этот алгоритм согласно инструкции должны применять все кассиры при выдаче сдачи: сначала выплачивается монетами максимально возможного номинала максимальная сумма, не превосходящая заданную, если осталась еще невыплаченная сумма, то к ней применяют ту же процедуру, только, конечно, уже с меньшим номиналом и т.д. В конце концов вся сумма будет выплачена.

Если его применить для решения задачи о точном булевом рюкзаке, на каждом шаге в рюкзак надо класть самый тяжелый предмет из тех, кто в него еще помещается. Если самый легкий предмет будет в него положен, и в рюкзаке еще останется место, то точного решения нет.

Но в общем случае эта задача является трудно решаемой.

Интересно, что булев рюкзак можно применять для передачи шифрованных сообщений по публичным каналам связи. Для этого используется описанная далее так называемая рюкзачная криптосистема.

Возьмем любую быстро растущую последовательность $a_1$, $a_2$, ..., т.е. такую, для которой всегда

$$a_1 + a_2 + \ldots + a_i < a_{i+1}.$$

Сообщение, которое мы хотим передать, например через интернет, сначала запишем в виде последовательности нулей и единиц $x_1$, ..., $x_n$ и вычислим сумму

$$N = a_1 x_1 + \ldots + a_n x_n.$$

Число N определяет последовательность $x_i$ однозначно, и если вам известно это число и последовательность $a_i$, то вы легко найдете последовательность $x_i$, применив жадный алгоритм. Значит, вместо этой последовательности вам достаточно послать это число $N$, и последовательность $a_i$, и получатель вашего письма быстро расшифрует сообщение. Но противник может перехватить сообщение и также его расшифровать. Поэтому последовательность $a_i$ передавать открытым текстом нельзя.

Тогда выберем число $M$, большее суммы всех $a_i$, и число $k$, $1 < k < M$, взаимно простое с $M$. Вычислим остатки от деления $ka_i$ на $M$ и обозначим их $b_i$.

Если числа $k$, $M$ велики и выбраны случайным образом, последовательность $b_i$ будет выглядеть как случайная. Вычислим

$$b_1 x_1 + \dots + b_n x_n.$$

Остаток R от деления этой суммы на $M$ будет такой же, как от деления на $M$ числа

$$kN = k a_1 x_1 + \dots + k a_n x_n.$$

Числа $M$, $k$ отправим сообщнику заранее и попросим их хранить в секрете. А вместо $a_i$ и N пошлем $b_i$ и $R$. Уравнение

$$b_1 x_1 + \dots + b_n x_n = R$$

в числах 0,1 имеет такое же решение, что и уравнение

$$N = a_1 x_1 + \dots + a_n x_n.$$

Других таких решений оно не имеет, потому что любое другое решение $y_i$ было бы решением уравнения

$$a_1 x_1 + \dots + a_n x_n = N,$$

потому, что

$$0 \le a_1 x_1 + \dots + a_n x_n < a_1 + \dots + a_n < M,$$

а это решение единственно.

Получатель вашего письма, зная $k$, $M$, вычислит $a_i$ и $N$ (они находятся с точностью до слагаемого, кратного $M$, но так как $a_i$ и

$$N < a_1 + \dots + a_n < M,$$

то на самом деле $a_i$ и $M$ определяются однозначно). Потом он найдет последовательность $x_i$.

А противник, зная эту криптосистему, но не зная $k$, $M$, вынужден будет или искать его полным перебором (что практически невозможно, если $M$ очень велико), или должен будет решать проблему рюкзака для неудобной системы $b_i$, т.е. решать $NP$-полную задачу.

Вначале подобные системы выглядели привлекательно, но потом в них были обнаружены слабости, и сейчас их признали ненадежными и вывели из употребления.

*Решите задачу о точном булевом рюкзаке для системы весов 67, 7, 14, 28, 56, 112, 97 и рюкзака вместимостью 346.*

## Упаковка в контейнеры

Одним из важных с прикладной точки зрения частных случаев общей задачи булевого программирования является задача об упаковке в контейнеры. Ее можно сформулировать следующим образом. Даны положительные числа $w_1, \dots, w_n$.

Их нужно разбить на наименьшее число множеств так, чтобы сумма слагаемых в каждом из них была не больше единицы (единица — это вместимость контейнера, задача состоит в минимизации числа контейнеров).

Оказывается, задача упаковки в контейнеры $NP$-полна.

Действительно, к ней можно полиномиально свести задачу разбиения. Пусть даны произвольные числа $w_1, \dots, w_n$, сумма которых равна $M$. Положим $W_i = 2w_i/M$, тогда $W_1 + \dots + W_n = 2$.

Решим задачу упаковки этих грузов в единичные контейнеры. Их нужно не менее двух. Два контейнера достаточно только в случае, если они будут полностью заполнены, т.е. грузы будут разбиты на две части так, что сумма весов $W_i$ из одной части равна 1, и сумма весов $W_i$ из другой части равна 1.

Но тогда сумма чисел $w_i$ с теми же индексами (номерами), что и у грузов $W_i$ первой части, равна $M/2$, и сумма оставшихся чисел $w_i$ тоже равна $M/2$, т.е. решена задача разбиение.

Обратно, если числа $w_i$ разбиты на две части, суммы которых равны, то они равны точно $M/2$, а соответствующие суммы весов грузов $W_i$=$2w_i/M$ будут равны 1 каждая, значит их можно упаковать в два контейнера, т.е. решена задача упаковки.

Поэтому задача упаковки в контейнеры трудно решаемая.

Но на практике не обязательно ее решать точно, обычно всех устраивает быстрое, хотя и приближенное решение.

Для этого можно применить следующий «пожарный алгоритм». Он очень прост: очередной предмет (выбранный случайно) помещается в первый же контейнер, в который он влезает.

Американский математик Джонсон доказал (доказательство длинное и поэтому здесь не приводится), что если число заполненных этим алгоритмом контейнеров обозначить $f$, а минимальное число контейнеров, в которые можно поместить данный набор предметов обозначить $m$, то

$$f \leq 1{,}7\, m + 2.$$

Другими словами, «пожарный алгоритм» решает задачу упаковки хуже, чем точный алгоритм не более чем в 1,7 раза. Более просто доказать, что погрешность этого алгоритма в сравнении с точным не более чем 100 процентов.

*Докажите для пожарного алгоритма неравенство $f < 2m$, где $m$ — число контейнеров при оптимальной упаковке.*

Указание. Пусть $s_i$ — загрузка $i$-го контейнера.

При пожарном алгоритме найдется не более, чем один контейнер, для которого $s_i < 1/2$ (если $s_i < 1/2$, значит все оставшиеся предметы имеют вес, больший $1/2$, иначе бы их кидали в $i$-й контейнер, поэтому при $j > i$ всегда $s_i \geq 1/2$), значит

$$f < 1 + 2(s_1 + \ldots + s_n).$$

Очевидно, что

$$m \geq s_1 + \ldots + s_n \ldots$$

Более эффективной оказалась следующая модификация пожарного алгоритма, в которой вначале все предметы упорядочиваются по весу (это можно сделать простейшими алгоритмами сортировки за $O(n^2)$ перестановок этих предметов, а наиболее быстрыми — за $O(n \log n)$ перестановок), а потом применяется пожарный алгоритм. Эту модификацию иногда называют «упаковкой при переездах» (когда есть время на сортировку).

Джонсон доказал, что число $t$ используемых контейнеров в этом алгоритме не больше $11m/9+4$, причем существуют задачи упаковки, для которых $t \geq 11m/9$ и $m$ сколь угодно велико.

Доказательство верхней оценки очень длинное, и поэтому здесь не приводится.

Более просто доказывается оценка $t \leq 3m/2$. Читатель может попытаться получить ее самостоятельно, а также решить следующую задачу.

*Пусть имеется 3 группы по $6m$ предметов с весами $1/2+\varepsilon$, $1/4+2\varepsilon$, $1/4+\varepsilon$, соответственно и еще $12m$ предметов с весами $1/4-2\varepsilon$. Покажите, что при достаточно малых $\varepsilon>0$ оптимальная упаковка требует $9m$ контейнеров ($6m$ контейнеров с предметами $1/2+\varepsilon$, $1/4-2\varepsilon$, $1/4+\varepsilon$, и $3m$ контейнеров с предметами $1/4-2\varepsilon$, $1/4+2\varepsilon$ по две штуки каждого), а алгоритм упаковки при переездах сначала заполнит $6m$ контейнеров наибольшими предметами $1/2+\varepsilon$, $1/4+2\varepsilon$, потом $2m$ контейнеров тройками предметов веса $1/4+\varepsilon$, а потом $3m$ контейнеров четверками оставшихся предметов $1/4-2\varepsilon$. В этом случае отношение числа контейнеров в этой упаковке к оптимальному числу равно $11/9$.*

## Жадность не всегда плохо

Приближенные алгоритмы упаковки в контейнеры, приведенные выше, относятся к классу так называемых жадных алгоритмов, также как и алгоритм точной упаковки булевого рюкзака в случае быстро растущей последовательности весов предметов.

Точное определение понятия жадный алгоритм дать затруднительно, но далее будут приведены еще некоторые их примеры. Было время, когда некоторые практики считали жадные алгоритмы точными и оптимальными. В упомянутом частном случае задачи о булевом рюкзаке это действительно так, но в разделе об упаковке в контейнеры приведены примеры ситуаций, когда жадный алгоритм упаковки при переездах давал, хотя и быстро, но лишь приближенный результат. Правда, рассматриваемая там задача была трудно решаемой. Приведем пример простой задачи, понятной и младшеклассникам, в которой жадный алгоритм не дает наилучшего результата.

*Семья из четырех человек хочет ночью перейти мост. Мост выдерживает двоих, а на всех только один фонарик. Отец переходит мост за 1 мин, мать — за две, старший сын — за пять, а младший — за десять. Как им перейти мост за минимальное время?*

Ответ: 17 минут.

Но в задачах упаковки рюкзака жадные алгоритмы не так уж и плохи.

Рассмотрим следующий жадный алгоритм булевой упаковки рюкзака. Пусть предмет $p_i$ имеет вес $w_i$ и стоимость $c_i$, $i=1, \ldots, n$. Формально задача состоит в нахождение булевых переменных $x_i$, таких, что суммарный вес

$$w_1x_1+\ldots+w_nx_n\le w,$$

и суммарная стоимость

$$c_1x_1+\ldots+c_nx_n$$

максимальна (эта задача иногда называется оптимизационный булев рюкзак). Вычислим удельные стоимости $c_i/w_i$ и упорядочим их в порядке убывания, пусть, например, нумерация выбрана так, что $c_1/w_1\ge\ldots\ge c_n/w_n$. Пусть $C$ — наибольшая из стоимостей.

Упаковываем предметы в рюкзак в порядке убывания удельной стоимости (в этом и проявляется жадность алгоритма) до тех пор, пока они в него вмещаются, т.е. вычисляем $k$, такое, что

$$w_1 + \ldots + w_k \leq W < w_1 + \ldots + w_{k+1}.$$

Если

$$C' = c_1 + \ldots + c_k \geq C,$$

то рюкзак уложен предметами $p_1, \ldots, p_k$. Если же $C' < C$, то кладем в рюкзак только один предмет $p_i$ максимальной стоимости $c_i = C$. Обозначим $C_ж$ стоимость уложенных этим алгоритмом предметов и сравним ее с (возможно, неизвестной нам) стоимостью оптимальной упаковки $C_{оп}$.

Можно доказать (но доказательство довольно длинное), что

$$C_ж \geq C_{оп} / 2,$$

т.е. жадная упаковка не более чем в два раза хуже оптимальной.

Доказано также, что для любого $\varepsilon > 0$ можно за полиномиальное время $N^{c_\varepsilon}$ найти булеву упаковку рюкзака со стоимостью, не меньшей $1 - \varepsilon$ от стоимости оптимальной упаковки.

Правда, показатель степени будет стремиться к бесконечности при стремлении $\varepsilon$ к нулю.

## Калейдоскоп Калейдоскоп Калейдоскоп

### Создана самая малая генетически кодируемая флуоресцентная метка

Флуоресцентные белки — важнейший инструмент в современных биологических исследованиях. При этом размер белка — одна из его ключевых характеристик: чем меньше флуоресцентная метка, тем слабее она влияет на поведение изучаемого объекта. Коллектив ученых из ИБХ им М. М. Шемякина и Ю. А. Овчинникова РАН и МФТИ разработал флуороген-активирующий белок nanoFAST размером всего 98 аминокислот, и это наименьший размер среди всех генетически кодируемых флуоресцентных меток. Работа выполнена при поддержке гранта РНФ и опубликована в престижном журнале Chemical Science.

Флуоресценция — способность веществ переизлучать поглощенный свет с увеличением его длины волны. Это свойство ряда молекул является чрезвычайно удобным для визуализации процессов, происходящих в живых системах: можно легко разделить потоки отражаемого и переизлучаемого света и выделить только те объекты, которые способны флуоресцировать. Флуоресцентные метки (молекулы, которые пришиваются к объекту, за которым требуются следить) очень широко используются в биологии для определения их локализации в клетке, для изучения взаимодействий между белками и прочих целей. Наиболее удобными метками являются белки — их можно генетически закодировать и легко пришить к объекту исследования с использованием подходов генной инженерии. Среди флуоресцентных меток можно выделить флуороген-активирующие белки. Такие белки работают в паре с небольшой органической молекулой — флуорогеном. Сами по себе ни белок, ни флуороген не светятся или светятся очень тускло, однако как только они провзаимодействуют между собой, их комплекс становится яркой флуоресцентной меткой.

*Источник: https://mipt.ru/news*