Лекция 3

ЗАДАЧИ КЛАССОВ P И NP

1. Сложные и простые алгоритмы

Сложность вычислительных алгоритмов обычно измеряется временем, необходимым для обработки входных данных. Если входными данными являются числа, то сложность алгоритма есть функция количества m десятичных цифр наибольшего из чисел.

Время выполнения *простых* алгоритмов яляется пропорциональной степени числа m или пропорциональной полиному от m. Например, перемножение двух m-значных чисел можно осуществить с помощью $O(m^2)$ алгоритма.

Алгоритм считается *сложным*, если время его выполнения растет быстрее, чем любая степень m. *Экспоненциальные* алгоритмы, для которых время выполнения растет как e^{am}, $a > 0$, считаются суперсложными и их выполнение считается крайне трудоемким.

1.1. Сложность алгоритма факторизации. Например, алгоритм факторизации числа n с помощью проверки делением на все числа, меньшие $\sqrt{n}$, является экспоненциальным: если число n состоит из m десятичных цифр, то необходимо осуществить $\sqrt{n} \approx 10^{m/2}$ или e^{am} проверок при $a = \frac{1}{2}\ln 10 \approx 1.15$.

Говорят, что алгоритм принадлежит классу **P**, если он выполняется за полиномиальное время (в таком случае мы также говорим, что алгоритм является полиномиальным). Все другие алгоритмы относятся к классу **NP**.

Прогресс в развитии математических методов имеет в рассматриваемых задачах намного более важное значение, чем увеличение быстродействия компьютеров. В 1977 году Рональд Ривест, один из авторов современной криптографии, писал, что для факторизации 125-значного числа с большими делителями необходимы 40 квадриллионов лет. Однако уже в 1994 году была осуществлена факторизация 129-значного числа.

Одним из признанных достижений компьютерных наук XXI столетия явилось доказательство, опубликованное в 2002 году тремя индийскими математиками Агарвалом, Каяном и Саксеном в статье под названием "*PRIMES is in* **P**". Основной результат этой работы можно предсказать по ее названию: авторы предложили полиномиальный алгоритм проверки чисел на простоту.

2. Как проверить пароль, не зная его?

Описанную идею сохранения шахматного кода можно использовать и для более "серьезных" задач. Например, при выдаче денег из банкомата центральный компьютер банка может проверить пароль клиента, не зная самого пароля! На первый взгляд и эта задача кажется неразрешимой, но теория сложности поможет и тут.

В банковской практике встречаются случаи, когда клиенты хранят ценные документы в специальных ячейках. Клиенты не желают, чтобы банк (или кто-либо другой) знал пароль доступа к ячейке. В силу важности документов пароль меняется после каждого использования ячейки. Можно ли скрыть пароль от банка, но при этом иметь доступ к своей ячейке?

Предположим, что пароль — это 100-значное простое число p (такой пароль представить себе трудно; поэтому предположение не вполне реалистично, но позволяет про-

ще рассказать о возможном решении). В тот момент, когда клиент выбирает пароль p, он выбирает также и другое простое число q, теперь 101-значное. Банку сообщается их произведение $N = pq$. В любой момент, когда клиент желает снять сумму со своего счета, он вводит пароль p, а компьютер банка проверяет является ли p делителем N. Проверка делимости одного числа на другое занимает считанные мгновения. Однако задачу нахождения делителей N можно решить только за экспоненциальное время, то есть практически никогда для 100-значных паролей. Это означает, что никто счетом клиента, кроме него самого, воспользоваться не может, даже зная число N, которое содержит всю информацию о пароле.

3. Еще одна экспоненциальная задача

Большое количество задач класса **NP** можно найти в теории графов. Среди них встречаются и такие задачи, что специалисты верят в их простоту, но в настоящее время не могут построить подходящий алгоритм. Одной из них, известной под именем задачи о сватовстве, можно придать форму занимательного рассказа.

3.1. Первая задача короля Артура. При дворе короля Артура было 150 неженатых рыцарей и 150 незамужних дам. Королю пришло на ум переженить их, но дело осложнялось тем, что не все из них были согласны с предлагаемыми королем вариантами. Король призвал своего советника Мерлина и под страхом казни приказал ему найти вариант, который устроил бы все пары.

Мерлин сразу осознал невозможность простого перебора всех 150! вариантов для решения задачи. Однако, имея фантастические таланты, он в назначенный срок смог во всем блеске продемонстрировать их перед своим королем. Мерлин попросил 56 отобранных им дам встать с одной

стороны королевского трона (принцип отбора знал только сам Мерлин); с другой стороны встали 95 отобранных Мерлином рыцарей. После этого королевский советник задал вопрос: “Желает ли кто-нибудь из 56 прекрасных дам выбрать себе в мужья одного из этих 95 достойных рыцарей?”. Когда дамы хором ответили “Нет!”, Мерлин обратился к Артуру: “Ваше величество, как можно найти мужей этим 56 дамам среди оставшихся 55 рыцарей?”.

Способ, которым Мерлин нашел комбинацию 56 дам и 95 рыцарей, остается загадкой и по сей день. Однако его рассуждения безупречны и абсолютно убедительны.

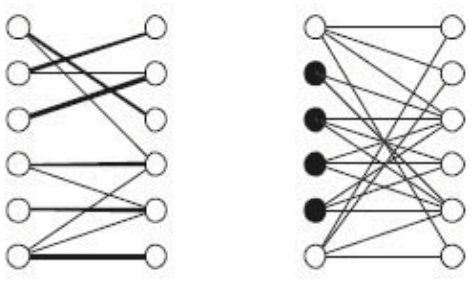

Рис. 1.
Два графа для задачи об
идеальном паросочетании

В этой задаче, как и в задаче о пароле для банка, проверка конкретного условия труда не представляет. Но нахождение самого этого условия является **NP** задачей!

Задача, которую мы только что рассмотрели, в теории графов называется задачей об *идеальном паросочетании.* На рис. 1 представлены два примера графов, которые можно использовать в этой задаче. Для левого графа задача имеет решение, для правого — нет.

Каждый отрезок на представленных графах обозначает допустимое паросочетание. Для левого графа решение задачи изображено жирными отрезками. Для правого графа

задача о паросочетании не имеет решения: 4 жирные точки соединены всего с тремя точками справа.

Описанная история при дворе короля Артура вскоре получила продолжение.

3.2. Вторая задача короля Артура. Через некоторое время король Артур стал замечать, что как ни рассади рыцарей за круглым обеденным столом, некоторые из соседей громко спорят и даже переходят на взаимные оскорбления. Впечатленный решением Мерлина в деле о сватовстве, король призвал его еще раз и приказал рассадить рыцарей так, чтобы раз и навсегда прекратить распри за столом. Мерлин моментально сообразил, что ни один из имеющихся 150! способов не приведет к успеху. Королю же он сказал следующее: “Ваше величество, Ваше желание прекратить надоедливые споры похвально в высшей степени: что может быть важнее мира и покоя при приеме пищи? Однако выполнить его невозможно ни при каких обстоятельствах. Если бы был такой рыцарь, у которого только один друг, невозможность решения была бы очевидной, поскольку каждый имеет двух соседей. Однако каждый из этих достойных рыцарей, Ваше величество, имеет много больше друзей, чем одного, и тем не менее миролюбивое расположение все же невозможно. Я могу доказать это, но мои объяснения будут слишком сложны и займут слишком много Вашего драгоценного времени. Они могут даже продолжаться и до конца жизни”. Не впечатленный такой перспективой, король Артур, поколебавшись немного, все же отказался от своей миротворческой затеи.

В теории графов вторую задачу короля Артура называют нахождением *гамильтонового пути*. Многие считают эту задачу полиномиальной, но алгоритм ее решения за полиномиальное время пока не известен.