Задание составлено на основе материалов большого числа моих коллег: Саши Иванова, Дани Селихановича, Сергея Шестакова, Алексея Балицкого, Рената Гимадеева и Ильи Козлова.

Рекомендации к чтению

- Кормен, 31 глава (теоретико-числовые алгоритмы).

Ключевые понятия: P, NPC, модулярная арифметика, КТО, RSA.

Обязательные задачи

Сначала две задачи на закрепление темы NP-полноты. Пункты первой сформулированы в духе задач с курсовых контрольных: верно ли что-то? В таких задачах нужно ответить да или нет и **доказать ответ**. Напомню, что для того чтобы доказать, что NP-язык является NP-полным, нужно привести сводимость **к нему** известного NP-полного языка, а для того чтобы показать, что NP-язык не является NP-полным, можно доказать, что он принадлежит P, тогда в предположении $P \neq NP$, он не может быть NP-полным (обычно в задачах предполагается именно этот способ доказательства не NP-полноты). Во второй задаче проверьте себя, что вы хорошо понимаете доказательства на графах.

Задача 1 $(1+1+1+1+1+3+1+3+3)$

(a) Верно ли, что язык описаний графов на n вершинах, правильно раскрашиваемых в $n-4$ цвета, является $\mathcal{NP}$-полным?

(b) Верно ли, что язык L выполнимых КНФ, для которых существует хотя бы два выполняющих набора, является NP-полным?

(c) Верно ли, что язык L тавтологичных ДНФ является полным в классе $co-NP$ (относительно полиномиальной сводимости)?

(d) Верно ли, что всякий регулярный язык принадлежит $co-NP$?

(e) Верно ли, что язык чисел n, у которых максимальный простой делитель не больше, чем $100\log_2^2 n$, принадлежит P?

(f) Верно ли, что язык графов, в которые можно добавить столько же ребер, сколько уже есть, так чтобы в полученном графе существовал гамильтонов путь, принадлежит NPC?

Перед тем как решать следующие пункты вспомните, что такое язык $PARTITION$ (см. конспект 4 семинара).

(g) Язык задается набором: целые числа n, a, b и массив из n целых чисел, каждое из которых равно либо a, либо b. При этом весь набор можно разбить на две непересекающиеся части, что суммы в обеих частях одинаковы. Верно ли, что он NP-полный?

(h) Язык задается набором: целое число n и массив из n целых чисел, каждое из которых может быть равно неотрицательным степеням двойки: 1, 2, 4 и так далее. При этом весь набор можно разбить на две непересекающиеся части, что суммы в обеих частях одинаковы. Верно ли, что он NP-полный?

(i) Язык задается набором: целое число n и массив из n целых чисел. При этом весь набор можно разбить на две непересекающиеся части так, что суммы в обеих частях различаются не более чем на 10. Верно ли, что он NP-полный?

Задача 2 (2) Подробно докажите сводимость языков CLIQUE, INDEPENDENT-SET и VERTEX-COVER друг к другу. Если вы это уже сделали в предыдущем задании, то тогда докажите любую сводимость, которую еще не доказывали, из конспекта 4 семинара.

Теперь ТЧ. Все задание можно разделить на две части. Делайн по следующим задачам – вечер 16 марта.

Задача 3 (1) Чему может быть равно $\sum_{i=1}^{m} i$ по модулю m?

Задача 4 (2) Предложите алгоритм проверки непустоты пересечения конечного семейства целочисленных арифметических прогрессий.

Задача 5 (2) Решите систему

$$\begin{cases} x \equiv 24 \mod 36 \\ x \equiv 45 \mod 54 \\ x \equiv 53 \mod 107 \end{cases}$$

Напомним вкратце схему RSA, которую мы обсуждали на семинаре. Эта асимметричная схема (для шифрования и дешифрования применяются разные процедуры) характеризуется следующими параметрами: $n = pq$, где p, q – различные большие простые числа; открытый ключ (e, n), где e взаимно просто с $\varphi(n) = (p-1)(q-1)$; секретный ключ (d, n), где d обратно к e по модулю $\varphi(n)$. Пусть x – остаток по модулю n. Тогда процедура шифрования сообщения x выглядит так: $P(x) = x^e \pmod n$, а процедура дешифрования сообщения y выглядит так: $S(y) = y^d \pmod n$. Криптоустойчивость схемы основана на предполагаемой сложности задачи *факторизации* (про язык FACTORING не доказана его принадлежность P, но не доказана и его NP-полнота). Число n всем известно, но непонятно, как по нему вычислить $\varphi(n)$, если нам неизвестно разложение n на множители.

Для того, чтобы протокол обмена шифрованными сообщениями работал, нужно, чтобы процедуры шифрования и дешифрования были обратны друг к другу. Поэтому нужно проверить, что $P(S(x)) = S(P(x)) = x^{ed} \equiv x \pmod{n}$. Для случая, когда x взаимно просто с n, это следует из теоремы Эйлера: $x^{\varphi(n)} \equiv 1 \pmod{n}$.

Задача 6 (2) Во введенных выше обозначениях докажите, что $x^{ed} \equiv x \pmod{n}$ для остатка x, не являющегося взаимно простым с n.

Систему RSA можно использовать не только для шифрования сообщения, но и для создания электронной подписи своего сообщения. Представьте, что Вы Боб и Вы хотите отправить Алисе некоторое сообщение, так чтобы Алиса, получив его, смогла убедиться в том, что это сообщение действительно от Вас, а не от кого-то еще. Попробуйте сами придумать протокол, который позволит это реализовать (или посмотрите на википедии). Следующая задача про то как можно атаковать систему RSA в этом варианте (это конечно не единственная разновидность атак).

Задача 7 (2) Вы Робин Олмост Гуд, который ворует деньги у богатых, а затем что-то с ними делает. Вы хотите, чтобы злой богач McDuck (сокращенно M) подписал своей электронной подписью сообщение x, переводящее более 9000 денег на Ваш счет.

Однако, очевидно, Вы не добьетесь результата, послав M сообщение x даже через поддельный адрес или фишинговый сайт. Однако, пусть (e, n) – открытый ключ M, а (d, n) – его секретный ключ. Возьмем случайное число r по модулю n и отправим M сообщение $y = r^e x \mod n$. Если сообщение y выглядит достаточно невинно (пожертвования для бездомных утят или реклама по увеличению клюва), M может согласиться подписать y своей подписью $s_y = y^d \mod n$.

Если M подпишет сообщение, то как по s_y и известным Вам данным получить правильную подпись на сообщения x?

Дополнительные задачи (можно сдавать в течение семестра)

Задача 8 (2) В конспекте 5 семинара я показал, как решать системы уравнений по взаимно простым модулям (см. док-во КТО). Поймите почему этот алгоритм полиномиальный. Теперь заметьте, что способ решения системы по **не** взаимно простым модулям, который я вам показал, включает в себя факторизацию этих модулей. Задача факторизации алгоритмически сложная. Поэтому показанный мной способ, удобный при решении игрушечных систем, не является эффективным. Ваша задача – предъявить полиномиальный алгоритм решения системы

$$\begin{cases} x \equiv a_1 \mod n_1 \\ \vdots \\ x \equiv a_m \mod n_m, \end{cases}$$

где числа $n_1, \dots, n_m$ не являются взаимно простыми.

Подсказка: постарайтесь вспомнить, что я говорил про то как можно рассмотреть сначала всего 2 из m уравнений и свести их к одному. Продолжая так делать можно прийти к решению. Вам нужно восстановить подробности и показать полиномиальность этого алгоритма.

Задача 9 (5) Если использовать один и тот же модуль n в разных протоколах (с разными e, d), то образуется уязвимость, так как пользователь протокола (Боб1) знает не только n и e, но и d. Оказывается, зная одну пару e, d можно легко найти закрытые ключи всех остальных пользователей протокола с тем же модулем (естественно зная их открытые ключи и n, но не зная p, q). Предложите эффективный алгоритм (полиномиальный от длины описания задачи), который решает эту задачу, докажите его корректность и оцените асимптотику.