Повторение

Вам предлагается освежить пройденный материал, решив следующие задачи:

1. (2б) Дано рекуррентное соотношение:

$$T(n) = \frac{n}{2} \cdot T(n-1) + 1, \quad T(1) = 1$$

Используя **дерево рекурсии**, оцените $T(n)$ как можно точнее. Ваша оценка $f(n)$ должна задаваться явной формулой и должна быть асимптотически эквивалентной $T(n)$, т.е. должна удовлетворять следующему равенству:

$$\lim_{n\to\infty} \frac{f(n)}{T(n)} = 1$$

2. (2б)

1. Про f, g известно, что $f(n) = O(n)$, $g(n) = \Omega(n)$. Следует ли из этого $f(n) \cdot g(n) = \Omega(n^2)$?
2. Пусть $f(n) \geq 0$, $g(n) \geq 0$. Верно ли, что либо $f(n) = O(g(n))$, либо $g(n) = O(f(n))$?

3. (2б) Найдите Θ-асимптотику рекуррентностей:

$$T(n) = 2018T\left(\frac{n}{2017}\right) + \log(n!),$$
$$T(n) = 4T\left(\frac{n}{2}\right) + O\left(\frac{n^2}{\log n}\right)$$

4. (2б) Найдите Θ-асимптотику рекуррентностей:

$$T(n) = T(n-1) + n\sqrt{n},$$
$$T(n) = \sqrt{n}T(\sqrt{n}) + n$$

5. (2б) Случайная величина $f(x)$ определена на наборе случайных событий $x_1, \ldots, x_m$, которым соответствуют вероятности $p_1, \ldots, p_m$ таким образом, что $f(x_i) = y_i$. Найдите значения параметров k и b таких что линейная модель $g(x) = kx + b$ наилучшим образом приближает $f(x)$ в смысле среднеквадратичного отклонения:

$$\mathbb{E}(f-g)^2 = \sum_{i=1}^{m} p_i(f(x_i) - g(x_i))^2 \to \min$$

6. (2б) $L \in \mathbf{co-NP}$. Верно ли, что любой язык $N \subset L$ принадлежит $\mathbf{co-NP}$?

7. (2б) Язык `EUCLID` состоит из троек (a, b, c) натуральных чисел таких что $\gcd(a, b) = c$. Язык `NOHAMPATH` состоит из кодировок графов, в которых нет гамильтонова цикла. Докажите или опровергните, что если `NOHAMPATH` полиномиально сводится к `EUCLID`, то $\mathbf{NP} = \mathbf{co-NP}$.

8. (2б) Будем говорить, что граф G содержит (n, k)-конфигурацию если в нём есть k вершинно непересекающихся клик из n вершин каждая, причём для каждой пары n-клик есть ребро, концы которого принадлежат каждой из клик пары.

Будет ли $\mathbf{NP}$-полным язык $\texttt{GAR} = \{(G, n, k) \mid G \text{ содержит } (n, k)\text{-конфигурацию}\}$?

9. (2б) Пусть $L \in \mathbf{co-NP}$. Покажите, что $L^* \in \mathbf{co-NP}$.

10. (2б) Верно ли, что существует такая функция $f : \mathbb{N} \to \mathbb{N}$, что $\forall\, c, d > 0$ выполнено:

$$f(n) = \omega(n^c), \ f(n) = o(2^{nd}),$$

т.е. $f(n)$ растет быстрее любого заданного полинома, но медленнее любой заданной экспоненты?

11. (2б) Пусть стало известно, что $\mathbf{NP-complete} \cap \mathbf{co-NP} \neq \varnothing$. Верно ли, что отсюда следует, что $\mathbf{NP} = \mathbf{co-NP}$?

Линейные рекуррентности в кольцах вычетов

В качестве модельной задачи к линейным рекуррентам рассмотрим задачу прошлого года:

12. (3б) Рассмотрим трехдиагональную $n \times n$ матрицу

$$A_n = \begin{vmatrix} 3 & 1 & 0 & \dots & 0 & 0 & 0 \\ 1 & 3 & 1 & \dots & 0 & 0 & 0 \\ 0 & 1 & 3 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 3 & 1 & 0 \\ 0 & 0 & 0 & \dots & 1 & 3 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & 3 \end{vmatrix}$$

На главной диагонали стоят тройки, а на двух соседних — единицы. Пусть $d_n = \det A_n$.

1. Найдите рекуррентное уравнение для детерминанта $d_n = \det A_n$.
2. Найдите асимптотику абсолютного значения d_n при $n \to \infty$.
3. Вычислите точное значение d_{2018} по модулю 11.

. . . . . .

Рассмотрим способы вычислять d_n алгоритмически. Для рекуррентного соотношения второго порядка можно указать матрицу A такую что возведением этой матрицы в степень можно получать d_n за полиномиальное время.

$$\begin{pmatrix} g_{n+1} \\ g_n \end{pmatrix} = \begin{pmatrix} a & b \\ 1 & 0 \end{pmatrix} \begin{pmatrix} g_n \\ g_{n-1} \end{pmatrix} = \begin{pmatrix} a & b \\ 1 & 0 \end{pmatrix}^n \begin{pmatrix} g_1 \\ g_0 \end{pmatrix}$$

13. (1б) Сформулируйте, как должна выглядеть аналогичная матрица для рекурренты:

$$F_n = a_1 F_{n-1} + \dots + a_k F_{n-k}$$

Вычислять степень можно двоичным возведением в степень за $O(\log n)$ рекурсивных вызовов.

Альтернатива – решить рекуррентность явно через характеристическое уравнение $\lambda^2 = a\lambda + b$. Но тогда в решении будет иррациональность $d = \sqrt{a^2 + 4b}$ и общий вид решения при $d \neq 0$ будет:

$$\lambda_1 = \frac{a + \sqrt{a^2 + 4b}}{2},\ \lambda_2 = \frac{a - \sqrt{a^2 + 4b}}{2},$$
$$d_n = C_1 \lambda_1^n + C_2 \lambda_2^n$$

Проводить такие вычисления в целых числах может быть неудобно. А что если нас просят вычислить не d_n, а $d_n \pmod p$? В таком случае нас могут ожидать приятные сюрпризы:

1. Уравнение $x^2 \equiv d^2 \pmod p$ может быть разрешимо даже если d — иррациональное число. Например, $4^2 \equiv 5 \pmod{11} \implies 4 \equiv \sqrt{5} \pmod{11}$. Таким образом, если бы мы хотели считать рекурренту с $d^2 = 5$ в кольце остатков по модулю 11, мы бы просто подставили 4 вместо $\sqrt{5}$ и получили бы корректное решение.
2. Последовательность d_n всегда будет периодичной по модулю, так как d_n определяется по d_{n-1} и d_{n-2}, а они принимают только конечные значения, таким образом, у d_n есть период не выше p^2. Более того, если d^2 является квадратичным вычетом, то для простого модуля p по малой теореме Ферма верно:

$$\lambda_i^n \equiv \lambda_i^{n \bmod (p-1)} \pmod p$$

Таким образом, можно существенно понизить степень, в которую нам нужно возводить λ_i.

14. (2б) Обоснуйте, что если уравнение $x^2 \equiv a \pmod p$ разрешимо, то мы можем подставить его решение вместо $\sqrt{a}$ в общую формулу решения линейной рекурренты и получить верный ответ.

Но что делать если уравнение не разрешимо? Это значит, что подходящего элемента x в нашем распоряжении нет. Решение в таком случае до смешного простое. Давайте рассмотрим расширение кольца, в котором мы работаем, которое содержит подходящий элемент x. Для этого мы можем просто рассматривать числа вида $a + bx$, где a и b — элементы исходного поля, а x — формальный символ, такой что $x^2 \equiv a \pmod p$.

15. (1б) Выпишите явную формулу для произведения чисел: $(u_1 + u_2x)(v_1 + v_2x)$. Результат должен находиться в том же расширении, то есть, иметь вид $(w_1 + w_2x)$.

16. (3б) Обоснуйте допустимость такой подстановки — т.е., что после формальной подстановки x в формулы и подсчёта $F_n = a_n + b_nx$ мы будем получать верный ответ, т.е., $b_n \equiv 0 \pmod p$.

1°. (3б) Сработает ли такой подход если вместо λ_1 и λ_2 подставлять формальный символ λ, удовлетворяющий уравнению $\lambda^2 = a\lambda + b$? Адаптируйте метод, чтобы такая замена вела к ответу.

1*. (1.5б) Вам даны наборы чисел $\{F_1, \ldots, F_k\}$ и $\{a_1, \ldots, a_k\}$. Рассмотрим рекурренту:

$$F_n = a_1F_{n-1} + \cdots + a_kF_{n-k}$$

Предложите алгоритм работающий за $O(k^2 \log n)$, вычисляющий $F_n \pmod m$.

Квадратичные вычеты

Будем называть число a квадратичным вычетом по модулю p если уравнение $x^2 \equiv a \pmod p$ разрешимо. Пусть p — простое число. Введём *символ Лежандра*:

$$\left(\frac{a}{p}\right) = a^{\frac{p-1}{2}} \pmod p$$

2°. (2б) Докажите, что:

1. Если $a \not\equiv 0 \pmod p$ – квадратичный вычет, то $\left(\frac{a}{p}\right) \equiv 1 \pmod p$
2. Если $a \not\equiv 0 \pmod p$ – квадратичный невычет, то $\left(\frac{a}{p}\right) \equiv -1 \pmod p$

Таким образом, символ Лежандра вводит простой критерий для проверки числа на то, является ли он квадратичным вычетом. По определению символ Лежандра мультипликативен по первому аргументу:

$$\left(\frac{a}{p}\right)\left(\frac{b}{p}\right) = \left(\frac{ab}{p}\right)$$

В частности, $\left(\frac{x^2}{p}\right)$ равно либо 0, либо 1.

Уравнения в целых числах

Алгоритм Евклида

Пусть нам даны числа a и b и мы хотим найти их наибольший общий делитель $\gcd(a, b)$. Заметим, что если d является общим делителем a и b, то это также делитель $a + b$ и $a - b$.

17. (1б) Обоснуйте строго, что $\gcd(a, b) = \gcd(a, b - a)$.

Исходя из этого можно рассмотреть следующий алгоритм:

```
function gcd_sub(a, b):
    if a > b:
        swap(a, b)
    if a = 0:
        return b
    else
        return gcd_sub(a, b - a)
```

То есть, мы вычитаем меньший элемент из большего пока один из них не станет равен 0. Для такого случая известно, что $\gcd(a, 0) = a$. Данная процедура является экспоненциальной по входу (пример: $a = 1,\ b = 2^n$).

Чтобы это исправить, заметим, что мы будем вычитать a из b пока a не станет больше b. В итоге мы получим новое число $b' = b - a \cdot d,\ b' < a$. По определению остатка от деления это значит, что $b' = b \pmod a$. Таким образом, можно рассмотреть новую процедуру:

```
function gcd_div(a, b):
    if a > b:
        swap(a, b)
    if a = 0:
        return b
    else
        return gcd_div(a, b mod a)
```

18. (1б) Покажите, что `gcd_div` работает за полиномиальное время от двоичной записи a и b.

Диофантовы уравнения от двух переменных

Заметим, что при рекурсивном вызове алгоритма Евклида, мы передаём аргументами линейные комбинации a и b. Из этого следует, что в итоге мы выразим $\gcd(a, b)$ как линейную комбинацию a и b. Отсюда следует, что с помощью алгоритма Евклида мы можем получить решение уравнения:

$$ax + by = \gcd(a, b)$$

Произвольное диофантово уравнение $ax + by = c$ совместно только если c делится на $\gcd(a, b)$, значит если у нас есть уравнение $ax + by = c$, мы можем вынести $g = \gcd(a, b, c)$ и получить новое:

$$a' = a/g,\ b' = b/g,\ c' = c/g$$

Теперь если $\gcd(a', b') = 1$, можно решить $a'x + b'y = 1$ и умножить его решение на c. Покажем, как использовать расширенный алгоритм Евклида для решения $ax + by = \gcd(a, b)$. Как уже было сказано, мы передаём аргументами рекурсии линейные комбинации исходных a и b. Будем теперь передавать, с какими коэффициентами они являются линейными комбинациями исходных a и b. Пусть $a = x_1a_0 + y_1b_0,\ b = x_2a_0 + y_2b_0$ и $a < b$, тогда $a' = a,\ b' = b - \lfloor b/a \rfloor \cdot a$, откуда получаем:

```
function gcd_ext(a, b, x1 = 1, y1 = 0, x2 = 0, y2 = 1):
    if a > b:
        swap(a, b)
        swap(x1, x2)
        swap(y1, y2)
    if a = 0:
        return {x2, y2}
    else
        d = ⌊b / a⌋
        return gcd_div(a, b - d * a, x1, y1, x2 - d * x1, y2 - d * y1)
```

19. (1б) Найдите любое решение уравнения $228x + 322y = 10$

Если мы захотим найти все решения уравнения, можно воспользоваться тем, что они представляют собой сумму частного решения и всех решений уравнения $ax + by = 0$.

20. (2б) Даны $a, b \in \mathbb{Z}$. Найдите все целые решения уравнения $ax + by = 0$.

С помощью полученного результата запишите общее решение прошлой задачи.

Характерное применение расширенного алгоритма Евклида – деление по модулю. Надо решить:

$$ax = b \pmod m$$

Можно переписать это уравнение в виде следующего дифовантова уравнения:

$$a \cdot x - d \cdot m = b$$

Которое уже решается с помощью алгоритма Евклида. Заметим, что такой метод в случае не простого модуля m гораздо лучше более интуитивной идеи нахождения обратного с помощью теоремы Эйлера, т.к. последняя работает только для обратимых элементов и требует вычисления функции Эйлера от модуля, что равносильно нахождению факторизации числа m.

Дискретное извлечение корня

Пусть мы хотим программно решить уравнение $x^2 \equiv a \pmod p$. Так как мы считаем, что модуль простой, у него должен быть первообразный корень. Рассмотрим случай $a \equiv 0 \pmod p$ отдельно, тогда если положить $a \equiv g^b \pmod p$ и $x = g^y \pmod p$, получим уравнение $2y \equiv b \pmod{p-1}$, которое можно решить расширенным алгоритмом Евклида. Но для этого нужно знать g и b.

2*. (1б) Предложите полиномиальный алгоритм, решающий задачу дискретного извлечения корня по простому модулю.

Поиск первообразного корня

Некоторые достижения в теории чисел указывают на то, что наименьший первообразный корень всегда ограничен сверху многочленом от записи числа p (точнее, если верна гипотеза Римана, то первообразный корень найдётся среди первых $\log^6 p$ элементов кольца вычетов по модулю p). Ранее уже был приведён алгоритм проверки на то, что число g является первообразным корнем $(p-1)$. Факторизовав число $(p-1)$ за $O(\sqrt{p})$ перебором делителей мы можем рассмотреть числа $g^{(p-1)/d} \pmod p$, ни одно из них не должно быть эквивалентно единице для $d \neq 1$, если это так, то g — первообразный корень.

Дискретное логарифмирование

Итак, мы знаем g, теперь нам нужно решить уравнение $g^b \equiv a \pmod p$. Для этого можно использовать *алгоритм Шенкса* (он же *baby-step-giant-step*). Пусть $r = \lceil \sqrt{p} \rceil$, любую степень b можно представить как $b = b_1 + b_2 r$, где b_1 и b_2 — числа от 0 до $r - 1$.

Суть алгоритма соответствует идее *meet-in-the-middle* разбиения переборной задачи на две крупные части. В данном случае мы сразу находим все числа вида $g^0, g^r, g^{2r}, \ldots, g^{r(r-1)}$ и записываем их в некоторый ассоциативный (например, хеш-таблицу). Затем мы перебираем числа вида $g^0, g^{-1}, g^{-2}, \ldots, g^{-(r-1)}$. Пусть сейчас мы рассматриваем число $c = g^{-k}$, посчитаем число $ac \equiv g^{(b_1-k)+b_2 r} \pmod p$. Если мы нашли $k = b_1$, то мы будем иметь $ac \equiv g^{b_2 r} \pmod p$, а значит, мы сможем найти число ac в нашем ассоциативном контейнере, так как все числа вида g^{kr} мы посчитали заранее и сохранили. Это решит задачу логарифмирования за $O(\sqrt{p})$.

3°. (3б) Покажите, что задача дискретного логарифмирования и факторизации равносильны по сложности — если вы умеете факторизовать за полиномиальное время, то вы также можете вычислять за полиномиальное время дискретный логарифм и наоборот.

Китайская теорема об остатках

Пусть n, m_1, m_2 – некоторые целые числа. При этом $\gcd(m_1, m_2) = 1$. Пусть нам известны остатки от деления n на эти числа: $r_1 = n \pmod{m_1}$ и $r_2 = n \pmod{m_2}$. Тогда мы можем однозначно восстановить остаток от деления на их произведение: $r = n \pmod{m_1 m_2}$. Верно это и в обратную сторону – r однозначно задаёт r_1 и r_2:

$$n \equiv r \pmod{m_1 m_2} \iff \begin{cases} n \equiv r_1 \pmod{m_1}, \\ n \equiv r_2 \pmod{m_2} \end{cases}$$

В одну сторону всё довольно просто: $r_1 = r \pmod{m_1}$, $r_2 = r \pmod{m_2}$.

21. (1б) Обоснуйте это.

Научимся находить r по данным r_1, r_2. Будем искать ответ в виде $r = a \cdot m_1 + b \cdot m_2$. Это позволит нам разделить переменные в уравнениях, т.к. если мы теперь возьмём остатки по модулям m_1 и m_2, то получим:

$$\begin{cases} r_1 \equiv b m_2 \pmod{m_1}, \\ r_2 \equiv a m_1 \pmod{m_2} \end{cases}$$

Такие уравнения мы научились решать в предыдущем разделе, кроме того, учитывая, что $\gcd(m_1, m_2) = 1$, мы можем записать решение в явном виде с помощью обратных элементов:

$$\begin{cases} a \equiv r_2 \cdot m_1^{-1} \pmod{m_2}, \\ b \equiv r_1 \cdot m_2^{-1} \pmod{m_1} \end{cases}$$

Обобщим эту задачу. Пусть теперь нам дан набор из n попарно взаимно простых модулей m_i. Нам нужно по набору остатков от деления на эти модули r_i восстановить остаток r от деления на $m = m_1 m_2 \dots m_n$. Мы хотим вновь представить ответ в виде некоторой линейной комбинации, чтобы для каждого m_i все слагаемые, кроме одного занулялись. Этого можно добиться, представив r в следующем виде:

$$r = \sum_{i=1}^{n} x_i \cdot (m/m_i)$$

Тогда мы получим систему уравнений вида $r_i \equiv x_i \cdot (m/m_i) \pmod{m_i}$. Обозначим $M_i = m/m_i$, тогда решением будет $x_i = r_i \cdot M_i^{-1} \pmod{m_i}$ и, соответственно:

$$r = \sum_{i=1}^{n} r_i M_i M_i^{-1} \pmod{m}$$

Где M_i^{-1} – обратный к M_i элемент по модулю m_i. Идеи китайской теоремы об остатках найдут своё отражение позже, когда мы рассмотрим интерполяцию многочленов.

Не взаимно простые модули

Вновь рассмотрим пару уравнений:

$$\begin{cases} n \equiv r_1 \pmod{m_1}, \\ n \equiv r_2 \pmod{m_2} \end{cases}$$

Однако теперь не будем допускать, что $\gcd(m_1, m_2) = 1$. Самый простой способ решить эту систему — свести её к случаю когда модули не взаимно простые. Для этого можно рассмотреть разбиения m_1 и m_2 на степени простых и перейти к системе уравнений только по этим степеням.

22. (2б) Решите систему сравнений:

$$\begin{cases} x \equiv 7 \pmod{24} \\ x \equiv 3 \pmod{28} \\ x \equiv 10 \pmod{21} \end{cases}$$

В общем же случае чтобы получить решение за полиномиальное время, эту систему можно рассмотреть как сугубо систему диофантовых линейных уравнений:

$$\begin{cases} n = r_1 + k_1 m_1 \\ n = r_2 + k_2 m_2 \end{cases} \iff k_1 m_1 - k_2 m_2 = r_2 - r_1$$

Таким образом, любую пару уравнений можно свести к линейному диофантовому уравнению от двух неизвестных. Найдя какое-то решение (k_1, k_2), мы также однозначно определим, чему равен x по модулю $\frac{m_1 m_2}{\gcd(m_1, m_2)}$, таким образом мы можем выбрасывать наши уравнения, пока у нас не останется лишь одно, которое и будет давать ответ.

Системы диофантовых линейных уравнений

Дана система из n уравнений с m неизвестными:

$$\begin{cases} a_{11}x_1 + \cdots + a_{1m}x_m = b_1, \\ a_{21}x_1 + \cdots + a_{2m}x_m = b_2, \\ \cdots\cdots\cdots\cdots\cdots\cdots, \\ a_{n1}x_1 + \cdots + a_{nm}x_m = b_n \end{cases}$$

Или в матричном виде $Ax = b$. При этом $a_{ij},\ b_i$ – целые числа и решение предполагается искать также в целых числах, то есть, перед нами система линейных диофантовых уравнений. Для её решения заметим, что следующие действия не влияют на совместность системы в силу их обратимости:

1. Прибавить строку j к строке $i \neq j$ с некоторым целым коэффициентом k.
2. Прибавить столбец j к столбцу $i \neq j$ с некоторым целым коэффициентом k.
3. Поменять две строки или два столбца местами.

При этом действия над строками реализуются умножением левой и правой частей уравнения на некоторую матрицу L, а действия над столбцами – умножением матрицы A на R справа и вектора переменных x на R^{-1} слева. С помощью данных преобразований можно алгоритмом Евклида получить в позиции a_{11} наибольший общий делитель всех элементов матрицы, а затем занулить все элементы в первой строке и первом столбце, кроме самого a_{11}. Затем процедура повторяется для вторых, третьих и т.д. строк и столбцов. В итоге получаем систему:

$$LAR(R^{-1}x) = Lb$$

При этом $D = LAR$ – диагональная матрица, в которой каждый следующий диагональный элемент делится на предыдущий. Представление матрицы $A = L^{-1}DR^{-1}$ называется нормальной формой Смита. В данном пространстве мы уже можем легко решить систему, которая будет состоять из уравнений вида $a'_{ii}x'_i = b'_i$. Эти уравнения решаются следующим образом:

1. $a'_{ii} = 0$:
   (a) $b'_i = 0$: Решением можно взять произвольное число $x'_i = C$.

(b) $b'_i \neq 0$: Система не совместна, т.к. получили $0 = b'_i \neq 0$.

2. $a'_{ii} \neq 0$:

(a) $b'_i = 0 \pmod{a'_{ii}}$: $x'_i = b'_i / a'_{ii}$.

(b) $b'_i \neq 0 \pmod{a'_{ii}}$: Система не совместна.

Наконец, получив $x' = R^{-1}x$ мы можем восстановить вектор решений x, умножив x' на R, которую мы могли получить при построении нормальной формы, для этого нужно было начав с $R = E$ дублировать на ней все проводимые преобразования столбцов.

4°. (3б) Дополните недостающие детали в описании алгоритма и оцените время его работы.

5°. (3б) Решите в целых числах систему уравнений в целых числах:

$$\begin{pmatrix} 8 & 27 & -5 \\ 1 & 5 & 7 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 1 \end{pmatrix}$$

Степенные башни

Научимся вычислять выражение $a_1^{a_2^{\cdot^{\cdot^{a_n}}}} \mod m$. Как и в случае с системами сравнений по не взаимно простым модулям у нас здесь есть два пути. Можно или считать это выражение по модулям всех степеней простых делителей m, где верна теорема Эйлера, или использовать возможно более простой и прямой метод описываемый следующим упражнением.

6°. (3б) Докажите следующее утверждение:

$$a^n \equiv a^{\varphi(m)+(n \bmod \varphi(m))} \pmod{m}$$

Для произвольных целых чисел a, m и $n \geq \log_2 m$.

Указание: Рассмотрите периодичность a^n по модулям p^d простых делителей m.

23. (2б) Вычислите $10^{25^{10}} \pmod{92}$.