

Алгоритмы и модели вычислений.

Домашнее задание № 11

Задача 0. Помимо очного семинара, запись предыдущего года я тоже посмотрел, так что вычисляю $\left(\frac{N}{41}\right)$, где $N = 13$ — мой номер в списке.

Решение.

Да не обязательно было, это я поступаю к контролю

$$\left(\frac{13}{41}\right) = (-1)^{\frac{41-1}{2} \cdot \frac{13-1}{2}} \cdot \left(\frac{41}{13}\right) = \left(\frac{41}{13}\right) = \left(\frac{2}{13}\right) = (-1)^{\frac{13^2-1}{8}} = (-1)^{\frac{168}{8}} = (-1)^{21} = -1$$

Задача 1. Имеются окрашенные прямоугольные таблички трёх типов: черный квадрат размера 2×2 , белый квадрат того же размера и серый прямоугольник 2×1 (последний можно поворачивать на 90°). Нужно подсчитать число способов $T(n)$ замостить полосу размера $2 \times n$. Найдите явную аналитическую формулу для $T(n)$ и вычислите $T(30000)$ по модулю 31.

Решение. Для нахождения рекурсивной формулы воспользуемся следующим наблюдением: так как размеры наших табличек не превышают 2×2 , то принципиально у нас есть 2 способа замостить край полосы:

1. замостить один крайний столбец: на это у нас есть всего один способ — поставить серый прямоугольник 2×1
2. замостить сразу два крайних столбца: на это у нас есть 3 способа — чёрный квадрат 2×2 , белый квадрат 2×2 , два серых прямоугольника 1×2 каждый (случай двух серых прямоугольников 2×1 покрывается предыдущим пунктом, поэтому тут не рассматривается)

Таким образом, получаем $T(n) = T(n-1) + 3T(n-2)$

Решим линейное рекуррентное соотношение: для начала запишем характеристическое уравнение

$$\lambda^2 = \lambda + 3 \Rightarrow \lambda_{1,2} = \frac{1 \pm \sqrt{13}}{2}$$

Таким образом $T(n) = C_1 \left(\frac{1 + \sqrt{13}}{2}\right)^n + C_2 \left(\frac{1 - \sqrt{13}}{2}\right)^n$, подберём константы из начальных условий: при $n = 0$ у нас есть 1 способ замостить ленту, при $n = 1$ — также 1 способ, то есть $T(0) = T(1) = 1$, тогда

$$C_1 + C_2 = 1 \quad C_1 \cdot \frac{1 + \sqrt{13}}{2} + C_2 \cdot \frac{1 - \sqrt{13}}{2} = 1$$

Решая, получаем $C_1 = \frac{1}{2} + \frac{1}{2\sqrt{13}} = \frac{1}{\sqrt{13}}\lambda_1$ и $C_2 = \frac{1}{2} - \frac{1}{2\sqrt{13}} = -\frac{1}{\sqrt{13}}\lambda_2$

И, окончательно:

$$T(n) = \frac{1}{\sqrt{13}} \cdot \left(\frac{1 + \sqrt{13}}{2}\right)^{n+1} - \frac{1}{\sqrt{13}} \cdot \left(\frac{1 - \sqrt{13}}{2}\right)^{n+1}$$

$$T(n) = T(n-1) + 3T(n-2) = 4T(n-2) + 3T(n-3) = 7T(n-3) + 12T(n-4) = 19T(n-4) + 21T(n-5) = 40T(n-5) + 57T(n-6) \equiv_{31} 9T(n-5) + 26T(n-6) = 35T(n-6) + 27T(n-7) \equiv_{31} 4T(n-6) + 27T(n-7) = 31T(n-7) + 12T(n-8) \equiv_{31} 12T(n-8)$$

Таким образом мы получили равенство $T(n) \equiv_{31} 12T(n-8)$. Тогда, так как $T(0) = 1 \equiv_{31} 1$, то

$$T(30000) \equiv_{31} 12T(30000-8) \equiv_{31} \dots \equiv_{31} 12^{\frac{30000}{8}} T(0) \equiv_{31} 12^{3750} = 12^{30 \cdot 125} = (12^{30})^{125} \overset{\text{по МТФ}}{\equiv_{31}} 1^{125} = 1$$

Итого получили, что $T(30000) \equiv_{31} 1$ *Эх, не хотите новые методы осваивать :-)*

Задача 2. Выполните задачи 1, Д-1 из приложенного файла (все по 1 баллу). **(+1)**

Решение. 1. (а) Посчитаем все пути длины 2, начинающиеся в вершине 1. Это пути $1 \rightarrow x \rightarrow y$, где $x \in \{1, 4\}, y \in \{1, 2, 3, 4\}$ и пути $1 \rightarrow z \rightarrow w$, где $z \in \{2, 3\}, w \in \{1, 4\}$. Их всего 12: $g(2) = 12$.

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \Rightarrow A^2 = \begin{pmatrix} 4 & 2 & 2 & 4 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 4 & 2 & 2 & 4 \end{pmatrix}$$

Заметим, что $g(1)$ совпадает с суммой элементов первой строки матрицы A^1 , а $g(2)$ совпадает с суммой элементов первой строки матрицы A^2 . Докажем по индукции, что на самом деле для любого $n \in \mathbb{N}$ верно, что $a_{ij}^{(n)}$ — элемент матрицы A^n , находящийся на i -й строке в j -м столбце, численно равен количеству путей из вершины i в вершину j , имеющих длину n .

- База очевидна: при $n = 1$ $A^n = A$ — матрица смежности графа, которая по определению является тем, что нам нужно
- Переход: пусть для $k - 1$ верно, тогда $a_{ij}^{(k)} = \sum a_{ir}^{(k-1)} \cdot a_{rj}$. В этой сумме пути $i \rightarrow r$ длины $k - 1$ для всех вершин r графа просто достраиваются до путей $i \rightarrow r \rightarrow j$, которые уже имеют длину k (если же пути $r \rightarrow j$ не существует, то $a_{rj} = 0$ и такие пути не будут учтены).

Таким образом, $g(n) = \sum_{r=1}^4 a_{1r}^{(n)}$ **(+1)**

(b) Обратимся к структуре нашей матрицы A и посмотрим, какие вообще элементы находятся в первой строке матрицы A^n .

- $a_{11}^{(n)} = a_{14}^{(n)} = g(n-1)$. Первое равенство верно в силу симметрии, а $a_{11}^{(n)} = g(n-1)$ следует из того, что в первом столбце матрицы A все элементы равны единице, поэтому $a_{11}^{(n)} = \sum_{r=1}^4 a_{1r}^{(n-1)} \cdot a_{r1} = \sum_{r=1}^4 a_{1r}^{(n-1)} = g(n-1)$.
- $a_{12}^{(n)} = a_{13}^{(n)} = 2g(n-2)$. Первое равенство верно в силу симметрии, а $a_{12}^{(n)} = 2g(n-2)$ следует из того, что во втором столбце матрицы A 1й и 4й элементы равны единице, в то время как 2й и 3й равны нулю, поэтому $a_{12}^{(n)} = \sum_{r=1}^4 a_{1r}^{(n-1)} \cdot a_{r2} = a_{11}^{(n-1)} + a_{14}^{(n-1)} = g(n-2) + g(n-2) = 2g(n-2)$.

Тогда, соответственно, сумма элементов первой строки матрицы A^n есть

$$g(n) = 2g(n-1) + 4g(n-2) \quad +1$$

(с) $g(n) \bmod 29 = (2 \cdot (g(n-1) \bmod 29) + 4 \cdot (g(n-2) \bmod 29)) \bmod 29$

На каждом шаге мы вычисляем только 3 арифметические операции (два умножения и одно сложение) и одно взятие остатка, это выполняется за линейное время. Также заметим, что так как $g(n-1) \bmod 29$ и $g(n-2) \bmod 29$ не превышают 29, а в памяти мы постоянно храним только их, то для этой задачи нам хватает константной памяти. Таким образом трудоёмкость процедуры есть $O(n)$ по времени и $O(1)$ по памяти. Конкретно при вычислении $g(20000) \bmod 29$ мы вычисляем приблизительно $20000 \cdot 4 = 80000$ операций $+1$

- (d) Для любого модуля m верно, что $g(n) \bmod m$ полностью определяется 2 числами: $g(n-1) \bmod m$ и $g(n-2) \bmod m$. Значит, всего различных вариантов $g(n) \bmod m \leq m^2$. Таким образом, на некотором шаге $T \leq m^2$ появится пара $g(T-1) \bmod m$ и $g(T-2) \bmod m$, ранее встречавшаяся в последовательности \Rightarrow будет вычислено $g(T)$, ранее встречавшееся и так далее, последовательность заиклиивается. Вернёмся к нашему модулю 29.

Сложность нахождения периода $O(1)$ как по времени, так и по памяти, так как нам точно достаточно просмотреть и сохранить в памяти 29^2 элементов последовательности.

Теперь для вычисления $g(n) \bmod 29$ нам достаточно вычислить $g(n \bmod T) \bmod 29$. Так что сначала вычисляем $n \bmod T$ за $O(\log n)$, затем просто достаём из памяти значение $g(n \bmod T) \bmod 29$ (которое мы положили в память при нахождении самого периода T). Таким образом, итоговая сложность есть $O(\log n)$ времени и $O(1)$ памяти. $+1$

2. (а) Решим линейное рекуррентное соотношение: для начала запишем характеристическое уравнение

$$\lambda^2 = 2\lambda + 4 \Rightarrow \lambda_{1,2} = 1 \pm \sqrt{5}$$

Таким образом $g(n) = C_1 (1 + \sqrt{5})^n + C_2 (1 - \sqrt{5})^n$, подберём константы из начальных условий: $g(0) = 1, g(1) = 4$. Тогда окончательно получаем:

$$g(n) = \frac{1}{\sqrt{10}} \cdot \left((5 + 3\sqrt{5})(1 + \sqrt{5})^n + (5 - 3\sqrt{5})(1 - \sqrt{5})^n \right)$$

- Найдём 10^{-1} по модулю 29: $10 \cdot 3 = 30 \equiv 1 \pmod{29}$, так что $10^{-1} = 3$
- Найдём $\sqrt{5}$ по модулю 29: $11^2 = 121 \equiv 5 \pmod{29}$, так что $\sqrt{5} \equiv 11 \pmod{29}$

Теперь получаем $g(n) = 3 \cdot (9 \cdot 12^n + 19^n)$

Также перед явным вычислением $g(n) \bmod 29$ необходимо заметить, что 29 — простое число, так что мы можем воспользоваться МТФ: $12^{28} \equiv 1 \pmod{29}$ и $19^{28} \equiv 1 \pmod{29}$, поэтому

$12^n \equiv 12^{n \bmod 28} \pmod{29}$, $19^n \equiv 19^{n \bmod 28} \pmod{29}$ и просто можем вычислить все эти 56 значений и положить в память (что займёт константное время и константную память).

Теперь для вычисления $g(n) \bmod 29$ осталось просто вычислить $n \bmod 28$ за $O(\log n)$, достать из памяти значения $12^{n \bmod 28} \bmod 29$, $19^{n \bmod 28} \bmod 29$, вычислить несколько арифметических операций и остаток результата по модулю 29 (результат не может превышать $3 \cdot (9 \cdot 28 + 28)$), так что взятие остатка от результата также есть константа.

Итоговая сложность есть $O(\log n)$ времени и $O(1)$ памяти.

$$A = g(20000) \equiv_{29} g(8) = 3 \cdot (9 \cdot 12^8 + 19^8) \equiv_{29} 3 \cdot (9 + 25) \equiv_{29} 3 \cdot 5 = 15$$

f1

- (b) Если же рассматривать другой модуль, в котором 5 не является квадратичным вычетом, то необходимо расширить наше поле таким образом, чтобы в нём $x^2 \equiv 5 \pmod{p}$ было разрешимо.

- Найдём 10^{-1} по модулю 23: $10 \cdot 7 = 70 \equiv_{23} 1$, так что $10^{-1} = 7$

Теперь получаем $g(n) = 7 \cdot ((5 + 3x)(1 + x)^n + (5 - 3x)(1 - x)^n)$

Так как $x^2 \equiv_{23} 5$, то в результате перемножения многочленов мы всё равно получаем многочлен первой степени.

Заметим, что так как мы живём в кольце вычетов по модулю 23, то перемножение многочленов первой степени занимает константное время. В результате же нам надо перемножить $2n$ многочленов первой степени, так что итоговая сложность алгоритма будет $O(n)$ по времени и $O(1)$ по памяти.

10,5

Задача 3. Верно ли, что существует такая функция $f: \mathbb{N} \rightarrow \mathbb{N}$, для любых констант $\forall c, d > 0$ выполнено

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

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

Решение.

Возьмём функцию $\tilde{f} = n^{\log n}$. Она, очевидно, растёт быстрее любого заданного полинома. Также она растёт медленнее любой заданной экспоненты (как такой предел считать я не знаю, но вольфрам подтверждает этот факт). Однако $\tilde{f}: \mathbb{N} \rightarrow \mathbb{R}$, так что возьмём округление вверх.

Итоговая функция $f = \lceil n^{\log n} \rceil$

10,5

Можу выдать справку на 1-ый курс, даушиться...

Задача 4. 1. Делится ли $4^{1356} - 9^{4824}$ на 35? Делится ли $5^{30000} - 6^{123456}$ на 31?

2. Найдите обратные $20 \pmod{79}$, $3 \pmod{62}$.
3. Найдите все решения уравнения $35x = 10 \pmod{50}$.
4. Имеет ли решение сравнение $x^2 = 1597 \pmod{3911}$

$e^{1/n}$ растёт медленнее e^{n^a} $\forall a > 1$
const

5. Найдите наименьшее натуральное число, имеющее остатки 2, 3, 1 от деления на 5, 13 и 7 соответственно.

Решение. 1. (а) Сразу посчитаем $\varphi(35) = 35 \cdot (1 - \frac{1}{5}) \cdot (1 - \frac{1}{7}) = 24$. Тогда, так как $(4, 35) = 1$ и $(9, 35) = 1$, можем пользоваться теоремой Лагранжа

$$4^{1356} - 9^{4824} = (4^{24})^{56} \cdot 4^{12} - (9^{24})^{201} \equiv_{35} 1 \cdot 4^{12} - 1 = 2^{24} - 1 \equiv_{35} 1 - 1 = 0$$

Так что $4^{1356} - 9^{4824}$ действительно делится на 35

- (б) 31 — простое число, так что будем пользоваться МТФ

$$5^{30000} - 6^{123456} = (5^{30})^{1000} - (6^{30})^{4115} \cdot 6^6 \equiv_{31} 1 - 1 \cdot 6^6 = 1 - 36^3 \equiv_{31} 1 - 5^3 = -124 \equiv_{31} 0$$

Так что $5^{30000} - 6^{123456}$ действительно делится на 31

2. (а) Легко заметить, что $20 \cdot 4 = 80 \equiv_{79} 1 \Rightarrow$ обратный к 20 по модулю 79 есть 4
 (б) Легко заметить, что $3 \cdot 21 = 63 \equiv_{62} 1 \Rightarrow$ обратный к 3 по модулю 62 есть 21
3. Для этого решим диофантово уравнение $35x + 50y = 10 \Leftrightarrow 7x + 10y = 2$.

$(7, 10) = 1$ так что решение в целых числах существует, для его нахождения воспользуемся расширенным алгоритмом Евклида

x	1	0	-1	3	6
y	0	1	1	-2	-4
7x+10y	7	10	3	1	2

Таким образом, $x = 6 + 10m, m \in \mathbb{Z}$ задаёт все решения уравнения $35x = 10 \pmod{50}$.

4. Посчитаем символ Лежандра $\left(\frac{1597}{3911}\right)$, если он будет равен 1, то 1591 — квадратичный вычет и решение есть, если же символ Лежандра окажется равным -1, то невычет и решений у сравнения нет.

$$\begin{aligned} \left(\frac{1597}{3911}\right) &= (-1)^{\frac{3910}{2} \cdot \frac{1596}{2}} \cdot \left(\frac{3911}{1597}\right) = \left(\frac{717}{1597}\right) = (-1)^{\frac{716}{2} \cdot \frac{1596}{2}} \cdot \left(\frac{1597}{717}\right) = \\ &= \left(\frac{163}{717}\right) = (-1)^{\frac{162}{2} \cdot \frac{716}{2}} \cdot \left(\frac{717}{163}\right) = \left(\frac{65}{163}\right) = (-1)^{\frac{64}{2} \cdot \frac{162}{2}} \cdot \left(\frac{163}{65}\right) = \\ &= \left(\frac{33}{65}\right) = (-1)^{\frac{32}{2} \cdot \frac{64}{2}} \cdot \left(\frac{65}{33}\right) = \left(\frac{-1}{33}\right) = (-1)^{\frac{32}{2}} = 1 \end{aligned}$$

$\left(\frac{1597}{3911}\right) = 1$, так что 1597 действительно является квадратичным вычетом по модулю 3911, а следовательно сравнение $x^2 = 1597 \pmod{3911}$ имеет решение

5. Воспользуемся КТО, чтобы найти множество целых чисел x , таких что
$$\begin{cases} x \equiv 2 \pmod{5} \\ x \equiv 3 \pmod{13} \\ x \equiv 1 \pmod{7} \end{cases}$$

Сначала найдём обратные элементы:

- по модулю 5: $91^{-1} = 1^{-1} = 1$
- по модулю 13: $35^{-1} = 9^{-1} = 3$, так как $9 \cdot 3 = 27 \equiv 1 \pmod{13}$
- по модулю 7: $65^{-1} = 2^{-1} = 4$, так как $2 \cdot 4 = 8 \equiv 1 \pmod{7}$

Тогда $x \equiv 2 \cdot 1 \cdot 91 + 3 \cdot 3 \cdot 35 + 1 \cdot 4 \cdot 65 = 757 \equiv 302 \pmod{455}$

Таким образом условию задачи удовлетворяют числа $x = 302 + 455n, n \in \mathbb{Z}$.

Наименьшее натуральное из них $m = 302$.

+2,5

Задача 5. Предложите полиномиальный алгоритм нахождения количества натуральных решений диофантова уравнения $ax + by = c$.

По какому параметру он полиномиальный?...

Решение. Если $d = (a, b)$ не делит c , то решений нет, что проверяется за $\text{poly}(|a| + |b|)$, если вычислять НОД с помощью алгоритма Евклида.

Если делит, то пользуемся расширенным алгоритмом Евклида, который также работает за полином от длины входа и выдаёт нам частное решение уравнения (x_0, y_0) , которое, в свою очередь, также полиномиально от длины входа.

Теперь мы можем сказать, что решением нашего уравнения является бесконечное подмножество целых чисел: $(x_0 + md, y_0 - md), m \in \mathbb{Z}$.

Чтобы определить количество решений в натуральных числах нам необходимо найти количество таких значений $m \in \mathbb{Z}$, что выполняется

$$\begin{cases} x_0 + md > 0 \\ y_0 - md > 0 \end{cases} \Rightarrow \begin{cases} m > -\frac{x_0}{d} \\ m < \frac{y_0}{d} \end{cases}$$

Таким образом мы вычисляем 2 значения $-\frac{x_0}{d}, \frac{y_0}{d}$, что происходит за $\text{poly}(|a| + |b| + |c|)$, так как все входящие значения занимают полином от длины входа. После чего выводим количество значений m , удовлетворяющих условиям из системы выше. Для этого достаточно взять модуль от разности двух выше вычисленных значений и округлить полученное значение вниз.

+1