

Второй курс, осенний семестр 2021/22
Практика по алгоритмам #10

Теория чисел

22 ноября

Собрано 20 декабря 2021 г. в 09:18

Содержание

| | |
|-------------------------------------|---|
| 1. Теория чисел | 1 |
| 2. Разбор задач практики | 3 |
| 3. Домашнее задание | 5 |
| 3.1. Обязательная часть | 5 |
| 3.2. Дополнительная часть | 6 |
| 4. Разбор домашнего задания | 7 |
| 4.1. Обязательная часть | 7 |
| 4.2. Дополнительная часть | 8 |

Теория чисел

1. Решето Эратосфена

- $n \leq 10^6$. Найти все простые числа на $[n^2, n^2 + n]$ за $\mathcal{O}(n \log \log n)$.
- Найти все простые на $[1, n]$ за $\mathcal{O}(n \log \log n)$ с $\mathcal{O}(\sqrt{n})$ памяти.

2. Применяем решето

Для каждого числа от 1 до n найти: сумму делителей и $\varphi(n)$.

3. Расширенный Евклид

- Вспомним/придумаем расширенного Евклида: найти $x, y: ax + by = \gcd(a, b)$.
- Доказать, что на каждом шаге Евклида $ax_i + by_i = \gcd(a, b)$ верно $\gcd(x_i, y_i) = 1$.
- Доказать, что на каждом шаге $|x_i| \leq |b|$ и $|y_i| \leq |a|$.
- Описать все решения диофантова уравнения $ax \equiv b \pmod{m}$.
- (*) Найти $x, y: ax + by = c, |x| + |y| \rightarrow \min$.
- (*) Придумайте нерекурсивную версию.

4. RSA

- Пусть $n = pq$, известно $\varphi(n)$, разложите n на множители.
- Пусть у нас есть «волшебный» оракул. Для любого открытого ключа (n, e) оракул может взломать 1% из всех возможных зашифрованных сообщений. Придумайте алгоритм, который взламывает любое сообщение. Матожидание времени работы $\mathcal{O}(\text{poly}(\log n))$.

5. RSA и реальный мир

Предложите быстрый способ передачи большого объёма закодированных данных. *Наш арсенал:* Диффи-Хеллман, RSA, хог-кодирование, псевдорандом.

6. Первообразный корень

Мы обсудили, как найти первообразный корень простого n . А если n не простое?

7. Биты сектора

Есть множество $A \subseteq [n]$ и k ячеек, каждая из которых умеет хранить $\mathcal{O}(\log n)$ бит информации. В каждый момент времени максимум t из k ячеек могут оказаться недоступны (мы можем узнать про ячейку, доступна ли она, и, если доступна, прочитать данные). Требуется организовать такой способ хранения информации, чтобы в любой момент времени можно было восстановить множество A .

- $k = (t + 1) \cdot |A|$.
- $k = t + |A|$ при $t = 1$.
- $k = t + |A|$.

8. Биномиальные коэффициенты по модулю

Дано m , предподсчёт за $\mathcal{O}(n)$, чтобы считать $\forall a, b < n \binom{a}{b} \pmod{m}$ за $\mathcal{O}(1)$.

9. Биномиальные коэффициенты по модулю

При данных $n \leq 10^6$ и простом $p \leq 10^6$ можно представить $n! = p^k r$. Поймите, как быстро найти k и $r \pmod{p}$. Пользуясь этим, найдите $\binom{n}{k} \pmod{m}$.

10. Башня степеней

Посчитайте $a_1^{a_2^{a_3^{\dots^{a_n}}}}$ mod m .

Пример: $m = 2^7 3^8$, $a = 2 \cdot 5$. В какую степень возводить?

Кто поможет!

11. (*) Квадратный корень по модулю

Хотим найти $r: r^2 \bmod p = n$. Представим $p = q2^s$.

Все равенства далее по модулю p .

a) Пусть у нас есть $r: r^2 = nt$, где $t^{2^k} = 1$. Основной шаг алгоритма в том, чтобы получить новые r и t , чтобы $r^2 = nt$ и $t^{2^{k-1}} = 1$.

Для этого шага воспользуемся величиной $c: c^{2^{s-1}} = -1$.

b) С чего начать и чем закончить алгоритм?

c) Где взять нужное c ?

d) Оценить время работы.

12. (*) Случайные квадратичные вычеты

Предложите алгоритм доказательства того, что x – квадратичный невычет по модулю n без разглашения.

Разбор задач практики

1. Решето Эратосфена

- За $\mathcal{O}(n \log \log n)$ нашли все простые от 1 до n . Теперь для каждого простого p_k вычеркнем все числа на отрезке $[n^2..n^2 + n]$, кратные ему. Сделаем это за $\mathcal{O}(\frac{n}{p_k})$.
- Предподсчитаем все простые от 1 до \sqrt{n} . Для каждого отрезка $[i\sqrt{n}..(i+1)\sqrt{n}]$ вычеркнем все непростые за $\mathcal{O}(\sqrt{n} + \sqrt{n})$ (количество простые + длина отрезка).

2. Применяем решето

```

1 int d[N]; // d[x] - минимальный простой делитель x
2 int cnt[N]; // cnt[x] - степень вхождения d[x]
3 int y[N]; // y[x] = x / d[x]^{cnt[x]}
4 // Запустили решето, нашли d[], d[1] = 0
5 cnt_div[1] = sum_div[1] = mu[1] = 1;
6 for (int x = 2; x < N; x++) {
7     int z = x / d[x];
8     if (d[x] == d[z])
9         cnt[x] = cnt[z] + 1, y[x] = y[z];
10    else
11        cnt[x] = 1, y[x] = z;
12    cnt_div[x] = cnt_div[y[x]] * (cnt[x] + 1);
13    sum_div[x] = sum_div[y[x]] * ((x / y[x] * d[x] - 1) / (d[x] - 1));
14    mu[x] = cnt[x] > 1 ? 0 : -mu[y[x]];
15 }

```

3. Расширенный Евклид

- Поддерживаем на каждом шаге представление $a_i = x_{a,i}a + y_{a,i}b, b_i = x_{b,i}a + y_{b,i}b$.
- Доказываем $(x_i, y_i) = 1$. Пользуемся тем, что алгоритм Евклида корректен. Если в какой-то момент $(x_i, y_i) = t$, то $|ax_i + by_i| \geq t \cdot \gcd(a, b) \Rightarrow t = 1$.
- Индукция. База: $b = 0 \Rightarrow x = 1, y = 0$. Переход:

$$r = a - kb = a \bmod b \wedge (a - kb)x' + by' = (a, b)$$

$$x = x' \wedge y = y' - kx', \text{ по индукции имеем } |x'| \leq b \wedge |y'| \leq r \Rightarrow |y' - kx'| \leq r + kb = a \quad \blacksquare$$

$$d) ax_0 + ty = (a, m) \Rightarrow ax_0 \frac{b}{(a,m)} \equiv b \bmod m \Rightarrow x \in \{x_0 \frac{b}{(a,m)} + k \frac{m}{(a,m)} \mid \forall k \in \mathbb{Z}\}.$$

$$e) \forall k \in \mathbb{Z} (x_0, y_0) + k(\frac{b}{\gcd(a,b)}, \frac{-a}{\gcd(a,b)}) - \text{решение. } |x_0 + k \frac{b}{\gcd(a,b)}| + |y_0 + k \frac{-a}{\gcd(a,b)}| -$$

почти линейная функция от k . Если бы мы знали, как раскрывается модуль, функция была бы линейной и минимум достигался бы на одном из краёв \Rightarrow

Решение: нарисуем прямую $ax + by = c$, она пересекает оси координат,

проверим два решения ближние к абциссе, два решения ближние к ординате.

4. Башня степеней

Факторизуем один раз m . $a^x \bmod m = \text{КТ0}(a^x \bmod p_i^{\alpha_i})$. Решаем задачу по модулю $p_i^{\alpha_i}$.

$$\text{Пусть } a = p_i^{\beta_i} g: (g, p_i) = 1 \Rightarrow a^x \bmod p_i^{\alpha_i} = [g^{x \bmod (p_i-1)p_i^{\alpha_i-1}} p_i^{\min(x\beta_i, \alpha_i)}] \bmod p_i^{\alpha_i}.$$

Мы свели задачу к двум: такой же и посчитать $\min(x, \frac{\alpha_i}{\beta_i})$.

5. RSA

- а) Пусть $n = pq$, известно $\varphi(n) \Rightarrow \varphi(n) = n - p - q + 1 \Rightarrow p + q = -(\varphi(n) - n - 1) \Rightarrow x^2 + (\varphi(n) - n - 1)x + n = 0$ имеет корни p, q .
- б) Нам дали $c = m^e \bmod n$. Загадаем случайное число r .
Дадим оракулу $r^e c \bmod n = (rm)^e \bmod n$, с вероятностью 0.01 он расшифрует.

6. Магия

Этот код считает обратные по модулю p ко всем $i = 1..p-1$. Возьмём $0 \equiv p = i \cdot \lfloor \frac{p}{i} \rfloor + (p \bmod i)$.
Домножим на $(p \bmod i)^{-1}$, получим $0 \equiv i \cdot \lfloor \frac{p}{i} \rfloor \cdot (p \bmod i)^{-1} + 1 \Rightarrow i \cdot -\lfloor \frac{p}{i} \rfloor \cdot (p \bmod i)^{-1} \equiv 1$.

7. Первообразный корень

Всё то же самое, только вместо $n - 1$ используем $\phi(n)$.

Напомним из алгебры, что первообразный корень есть только у чисел вида $2, 4, p^k, 2p^k$.

8. Биномиальные коэффициенты по модулю

e-maxx.

Если умеем по модулю p^k , то по КТО умеем и по модулю m .

9. (*) Квадратный корень по модулю

Алгоритм Tonelli-Shanks.

Домашнее задание

3.1. Обязательная часть

1. (2) Не самая наивная факторизация

Приходят запросы факторизовать число $a_i \leq N$.

Предсчет $\mathcal{O}(\sqrt{N})$, ответ на запрос за $\mathcal{O}(\sqrt{N}/\log N)$.

Ну, мы же вроде знаем, сколько бывает простых чисел от 1 до N ?

2. (2) Евклид и полезность

Лягушонок Вася живёт на прямой. Он умеет целеустремлённо прыгать только вперёд.

Устройство лапок позволяет прыгать ему только на a или b вперёд.

Может ли он попасть в точку c ?

3. (3) Я теряю корни

На лекции научились по данным p, a, b искать $x: x^a = b \pmod{p}$ (p простое).

Найти не один, а все такие x (заодно можно заметить, что на лекции был маленький обман).

4. (3) Эратосфен и гладкость

Обозначим i -е по возрастанию простое число, как p_i .

Назовём число b -гладким, если все его простые делители не превосходят p_b .

Дано $n \leq 10^6$. Для каждого $b \leq n$ найдите количество b -гладких чисел от 1 до n .

5. (3) Взлом RSA при малом e

Аня решила послать приглашение на секретную вечеринку Боре, Ване и Гоше. Аня разослала им одинаковый текст приглашения M закодированный с помощью RSA . У Вани, Бори и Гоши выбраны различные n , но ключ e у всех одинаковый: $e = 3$. Придумайте, как Дима сможет узнать (за полиномиальное от суммы длин всех чисел время), где будет происходить секретная вечеринка, если ему доступны все три шифрованных приглашения и открытые ключи.

3.2. Дополнительная часть

1. (3) Подсчёт d^p в лоб

Мы умеем за $\mathcal{O}(n)$ для всех $x \in [2, n]$ искать $d[x]$ – минимальный простой делитель x . Умеем искать и $p[x]$, степень вхождения $d[x]$ в x , за $\mathcal{O}(n)$ динамикой.

Докажите, что наивный поиск $p[x]$ работает тоже за $\mathcal{O}(n)$:

```
for (x = 2..N) for (y = x; d[y] == d[x]; y /= d[x]) p[x]++
```

Есть короткое док-во совсем без алгебры.

2. (4) $\pi(n)$ для больших n

$\pi(n)$ – количество простых от 1 до n .

С помощью решета Эратосфена умеем считать $\pi(n)$ за $\mathcal{O}(n)$. Можно быстрее. Эта задача даёт вам подсказку и предлагает придумать алгоритм.

$\Phi(n, d)$ – количество чисел от 1 до n , все простые делители которых больше d .

Можно вывести рекуррентную формулу пересчёта $\Phi(n, p_i)$, выразить через эту величину π , а из этого получить алгоритм.

Баллы будут ставиться за любое решение за $\mathcal{O}(n^{1-\epsilon})$. Существует решение за $\mathcal{O}(n^{2/3})$.

3. (5) $\binom{n}{k} \bmod m$

Рассмотрим алгоритм подсчёта $\binom{n}{k} \bmod m$ за $\mathcal{O}(n \log m) + \text{ФАСТ}(m)$. Разложим $m = \prod_i p_i^{\alpha_i}$.

$\binom{n}{k} = \frac{n!}{k!(n-k)!} \equiv \frac{f_p(n)}{f_p(k)f_p(n-k)} p^{cnt_p(n) - cnt_p(k) - cnt_p(n-k)} \bmod p^\alpha$. Далее используем КТО.

Слабое место этого алгоритма – факторизация. Придумайте аналог за $\mathcal{O}(\text{poly}(n, \log m))$.

4. (5) Быстрое дискретное логарифмирование

Придумайте любой алгоритм для поиска дискретного логарифма за $\mathcal{O}(n^{1/4} \cdot \text{poly}(\log))$.

Разбор домашнего задания

4.1. Обязательная часть

1. Не самая наивная факторизация

Предподсчитаем все простые от 1 до $N^{1/2}$. Их будет $\Omega(N/\log N)$.
Теперь при разложении на простые делить можно только на них.

2. Евклид и полезность

Во-первых, c должно делиться на \gcd . Если делится, получили уравнение $ax + by = c$ и решили. Осталась проблема с $x < 0, y < 0$. У нас есть весь класс решений: $x + k(b/g)$ и $y - k(a/g)$. Сдвинем за $\mathcal{O}(1)$ на $\langle b/g, -a/g \rangle$ нужное число раз в нужную сторону.

3. Я теряю корни

Ищем первообразный g . Логарифмируем b по g : $b = g^c$. Ищем x в форме $x = g^y$.

$$x^a = b \pmod p \Leftrightarrow (g^y)^a = g^c \pmod p \Leftrightarrow ya = c \pmod{p-1}$$

Если $c_1 = \gcd(c, p-1)$ не делится на $a_1 = \gcd(a, p-1)$, решений нет. Иначе решения есть.

$$c \neq 0 \Rightarrow y = \frac{c}{a_1} \cdot \left(\frac{a}{a_1}\right)^{-1} \pmod{\frac{p-1}{a_1}} \Rightarrow y = \frac{c}{a_1} \cdot \left(\frac{a}{a_1}\right)^{-1} + z \frac{p-1}{a_1} \pmod{p-1}, 0 \leq z < a_1 \quad (a_1 \text{ решений}).$$

Можно подумать, что $c = 0$ и $a = 0$ – отдельные случаи, но $\gcd(0, p-1) = p-1$ и по общей формуле всё правильно: $c = 0 \Rightarrow$ решаем $x^a = 1$. Если $a = 0, x$ любое, иначе $y = 0 \Rightarrow x = 1$.

4. Эратосфен и гладкость

Сперва за $\mathcal{O}(n)$ решетом посчитаем $\text{minPrimeDivisor}[i] \forall i = 1..n$. Затем за $\mathcal{O}(n)$ насчитаем динамику $\text{maxPrimeDivisor}[i] \forall i = 1..n$. $\forall i = 1..n$ сделаем прибавление на суффиксе за $\mathcal{O}(1)$.

5. Взлом RSA при малом e

$$m^3 = a_1 \pmod{n_1}$$

$$m^3 = a_2 \pmod{n_2}$$

$$m^3 = a_3 \pmod{n_3}$$

\Rightarrow КТО даёт $m^3 \pmod{(n_1 n_2 n_3)}$, в предположении, что у n_1, n_2, n_3 нет одинаковых простых делителей. Поскольку $m^3 < n_1 n_2 n_3$, знаем число m^3 , извлекаем корень.

4.2. Дополнительная часть

1. Подсчёт d^p в лоб

Эмпирическое наблюдение: $\lim_{N \rightarrow \infty} \frac{\sum p[x]}{N} = 1.6125 \dots$

Решение #1.

Зафиксируем $d[x] = y$.

i -й шаг внутреннего **for** мы сделаем только для $x: d[x] = y \wedge p[x] \geq i$, все такие x как минимум кратны $y^i \Rightarrow$ их не более $\frac{N}{y^i}$.

Для каждого x мы сделаем минимум 1 шаг, учтём его отдельно.

Обозначим \mathbb{P} – мн-во простых чисел. Оценим общее время работы T :

$$T \leq N + \sum_{y \in \mathbb{P}} \sum_{i \geq 2} \frac{N}{y^i} = N \left(1 + \sum_{y \in \mathbb{P}} \frac{1}{y^2} \cdot \frac{y}{y-1} \right) \leq N \left(1 + 2 \sum_{y \in \mathbb{P}} \frac{1}{y^2} \right) \leq N \left(1 + 2 \frac{\pi^2}{6} \right)$$

Чтобы сделать последний переход, мы решили, что считать сумму по всем простым слишком сложно, и посчитали просто по всем.

Решение #2.

?

2. $\pi(n)$ для больших n

Будет на доплекции.

3. $\binom{n}{k} \bmod m$.

Почему бы просто не посчитать $n(n-1) \dots (n-k+1)$, $k! \bmod m$ и не поделить?

Только потому что делить на 0 нельзя.

Если $\gcd(k!, m) = 1$, то $\binom{n}{k} \bmod m$ считается за $\mathcal{O}(n + \log m)$ (**for** и Евклид для деления).

Решение. $P = \{\text{все простые делители } k!\}$. $\forall x \in P \ x \leq k \leq n \Rightarrow$

За $\mathcal{O}(n)$ выделяем из m все простые до n , оставшаяся часть взаимнопроста с $k!$. Получили частичную факторизацию m . Далее решаем как в условии. Время по-прежнему $\mathcal{O}(n \log m)$.

4. Быстрое дискретное логарифмирование

?