Обозначения

$\sum\limits_{i=1}^{n} a_i = a_1 + a_2 + \ldots + a_n$

$\prod\limits_{i=1}^{n} a_i = a_1 \cdot a_2 \cdot \ldots \cdot a_n$

$f(n) = \mathcal{O}(g(n))$ — существуют такие числа a и b, что для всех n: $f(n) \le a + bg(n)$.

$f(n) = \Omega(g(n))$ — то же самое, что и $g(n) = \mathcal{O}(f(n))$.

$f(n) = \Theta(g(n))$ — одновременно $f(n) = \mathcal{O}(n)$ и $f(n) = \Omega(n)$.

$a \equiv b \pmod m$ — a и b имеют одинаковые остатки при делении на m.

Задачи

Задачи разбиты на блоки, которые отделены друг от друга линиями вида «— — —». На решение задач из каждого блока отводится некоторое время, после которого проводится их разбор. Рекомендуется сдавать задачи именно из текущего блока, если вы, конечно, уже не решили их все.

Задача 1. Предложите алгоритм вычисления НОД двух натуральных чисел за $\mathcal{O}(n^2)$, где n — ограничение на длину чисел в какой-то фиксированной системе счисления.

Задача 2. Даны n целых чисел, по модулю не превосходящих C. Рассмотрим следующий алгоритм вычисления их НОД.

```
answer = 0
for i = 0 ... n - 1:
    answer = gcd(answer, a[i])
```

Здесь функция `gcd` реализует алгоритм Евклида, использующий операции деления с остатком. Дайте асимптотическую оценку сложности работы данного алгоритма.

Задача 3. Пусть a, b, c — целые числа. Рассмотрим уравнение $ax + by = c$ относительно целых x, y.

a. Покажите, что, если c не делится на $gcd(a, b)$, решений нет.

b. Покажите, что при $c = gcd(a, b)$, решение есть.

c. Покажите, что решение существует тогда и только тогда, когда c делится на $gcd(a, b)$.

d. Покажите, что, если существует хотя бы одно решение, существует бесконечно много решений. Опишите их все.

Определение. Числа a и b называются взаимно обратными по модулю m, если $a \cdot b \equiv 1 \pmod m$.

Задача 4. Докажите, что $\frac{1}{1} + \frac{1}{2} + \ldots + \frac{1}{n} = \Theta(\log n)$.

Задача 5. Пусть $\tau(n)$ — количество натуральных делителей n. Докажите, что $\sum\limits_{i=1}^{n} \tau(i) = \mathcal{O}(n \log n)$.

Определение. Функция Эйлера $\varphi(n)$ — количество натуральных чисел меньших n, взаимно простых с n.

Задача 6. **a.** Выведите формулу для $\varphi(p^k)$ (p — простое число), которая будет зависеть от p и k.

b. Предложите алгоритм вычисления $\varphi(n)$ для произвольного n за $\mathcal{O}(\sqrt{n})$.

Определение. (*Китайская теорема об остатках*) Для любых n взаимно простых чисел $a_1, a_2, \ldots, a_n$ ($a_i \le C$) и n неотрицательных чисел $r_1, r_2, \ldots, r_n$ существует единственное число x ($0 \le x < \prod\limits_{i=1}^{n} a_i$) такое, что $x \equiv r_i \pmod{a_i}$ для всех $i = 1 \ldots n$.

Задача 7. В предположении, что все арифметические операции с числами любой длины выполняются за $\mathcal{O}(1)$, научитесь находить число x из предыдущего определения за $\mathcal{O}(n \log C)$.

Определение. (*Теорема Эйлера*) Для любого числа k, взаимно простого с n, $k^{\varphi(n)} \equiv 1 \pmod n$.

Частный случай этой теоремы при простом n называется *малой теоремой Ферма*. Этот случай примечателен тем, что $\varphi(p) = p - 1$ для простого p, отчего теорема принимает совсем простой вид.

Задача 8. Дано число a. Предложите алгоритм, который за $\mathcal{O}(\log m)$ вычислит обратный ему по модулю m или определит, что такого не существует, в предположении что:

a. m — простое.

b. m — не обязательно простое.

Задача 9. Число x называется первообразным корнем единицы степени k по модулю m, если $x^k \equiv m \pmod p$ и для любых $1 \le a \ne b \le k$ верно что $x^a \ne x^b \pmod m$. Для данных k и m найдите любой первообразный корень из 1 степени k по модулю m за время $O(m \log^2 m)$

— — —

Задача 10. Предложите способ ускорить решето Эратосфена, чтобы оно работало за линейное время.

Подсказка. Для каждого числа попробуйте найти минимальный простой делитель.

Задача 11. За $\mathcal{O}(n)$ для всех чисел от 1 до n найдите:

a. В какой степени минимальный простой делитель входит в его разложение.
b. Количество его простых делителей.
c. Количество его делителей.
d. Сумму его делителей.
e. Функцию Эйлера от него.

Задача 12. Научитесь вычислять $a \cdot b$ для натуральных a и b, используя только сложение, деление на 2 (в том числе с остатком), а также проверку на равенство 1 за $\mathcal{O}(\log a)$ операций сложения.

Задача 13. Для всех $k \le n \le X$ научитесь находить C_n^k по простому модулю $p > X$ за $\mathcal{O}(1)$, с предподсчётом за $\mathcal{O}(n)$.

Задача 14. (*Дискретное логарифмирование*) $a^x \equiv b \pmod m$, a и m взаимнопросты. Найти решение или определить, что его не существует, за время $\mathcal{O}(\sqrt{m} \log m)$.

Подсказка. Представьте x в виде $ky - r$ для $k = \lfloor \sqrt{m} \rfloor$.

Задача 15. Пусть вам дан простой модуль p. Вы можете сделать предподсчёт за $O(p \log^3 p)$, после чего требуется в онлайне отвечать на запрос «Для данного a и k найти все возможные x, что $x^k \equiv a \pmod m$» за время $O(\sqrt{p} + ans)$, где ans — размер ответа.

— — —

Задача 16. Найти количество натуральных чисел, меньших n, взаимно простых с числом m, за $\mathcal{O}(\sqrt{m})$

Задача 17. Найти сумму НОД по всем подотрезкам массива натуральных чисел, не больших C, за $\mathcal{O}(n \log C)$.

Задача 18. Найти сумму НОД по всем непустым подмножествам массива из n натуральных чисел, не больших C, за $\mathcal{O}(n + C \log C)$. Ответ найдите по модулю $10^9 + 7$.

Задача 19. Дан массив из n натуральных чисел, не больших C. Выпишем gcd по всем непустым подмножествам этого массива. Найти медиану выписанных чисел за $\mathcal{O}(n \cdot C \log C)$.

Задача 20. Назовём натуральное число кубастым, если его можно представить в виде $a^3 \cdot b$ для каких-то натуральных $a > 1, b \ge 1$. Найти количество кубастых чисел, не больших n. (n до 10^{18}, стандартный компьютер со стандартными ограничениями)