Билет 6. Решето Эратосфена

Определение

Решето Эратосфена — алгоритм для нахождения всех простых чисел до заданного целого N.

Алгоритм

Шаги алгоритма:

1. Выписываем числа $1, 2, \ldots, N$
2. Первое, большее 1 число — 2. Оно простое
3. Вычёркиваем все кратные 2 до N
4. Следующее невычеркнутое число — 3. Оно простое, вычёркиваем все кратные 3
5. Продолжаем пока не обработаем все числа до $\sqrt{N}$

Доказательство корректности

Корректность:
Когда вычеркнуты все числа, кратные простым, меньшим p, то все невычеркнутые числа, меньшие p^2, будут простыми.
Доказательство: Всякое составное $a < p^2$ уже вычеркнуто как кратное его наименьшего простого делителя, который $\leq \sqrt{a} < p$.

Важные замечания:

1. При вычёркивании кратных простого p начинаем с p^2
2. Алгоритм завершается, когда обработаны все простые $\leq \sqrt{N}$
3. Для каждого простого p вычёркиваем, начиная с p^2

Асимптотическая сложность

Асимптотическая сложность: $O(n \ln \ln n)$
Б/Д — означает "без доказательства" (доказательство сложности опущено)

Линейное решето Эратосфена

Определение

Проблема классического решета: Число может быть помечено как составное несколько раз (по каждому простому делителю).
Цель линейного решета: Каждое составное число помечается ровно один раз.

Алгоритм

Идея алгоритма:

- $p[x]$ — минимальный простой делитель числа x
- Любое число x можно представить как $x = p[x] \cdot i$
- i не имеет делителей меньших $p[x]$ (т.е. $p[x] \leq p[i]$)
- Перебираем i и для каждого перебираем простые $p \leq p[i]$

Доказательство корректности

Корректность:
Каждое составное число x помечается ровно один раз при $i = \frac{x}{p[x]}$ и $p = p[x]$.
Условие $p \leq p[i]$ гарантирует, что p — минимальный простой делитель x.

Процесс пометки в линейном решете

1	2	3	4	5	6	7	8	9	10
$p = 2$	$p = 2$	$p = 3$	$p = 2$	$p = 5$	$p = 2$	$p = 7$	$p = 2$	$p = 3$	$p = 2$

Простое Простое Составное

Каждое число обрабатывается ровно один раз!

Асимптотическая сложность

Асимптотическая сложность: $O(n)$
Обоснование:

- Внешний цикл: n итераций
- Внутренний цикл: для каждого i перебираем простые $\leq p[i]$
- Каждое составное число помечается ровно один раз
- Суммарное количество операций: $O(n)$

Объяснение ключевых моментов кода

- **Внешний цикл:** Перебираем все числа от 2 до n
- **Добавление простых:** Если i простое — добавляем в список primes
- **Внутренний цикл:** Для каждого простого p в списке:
  - Помечаем $i \cdot p$ как составное
  - Прерываем цикл если $i \cdot p > n$
  - **Ключевое условие:** Если p делит i — прерываем цикл

Почему прерываем при $i\%p == 0$?

$i = 4,\ p = 2$	Если продолжим:
$4\%2 == 0$	$4 \cdot 3 = 12$
Прерываем! $\longrightarrow$	Но $p[12] = 2 \neq 3$!

Нарушится условие минимальности

Преимущества линейного решета:

- **Линейная сложность** — $O(n)$
- **Однократная пометка** — каждое число обрабатывается один раз
- **Побочный продукт** — получаем массив минимальных простых делителей

Недостатки:

- Большая константа в асимптотике
- На практике классическое решето часто быстрее для больших n

```java
public class LinearSieve {

    public static List<Integer> linearSieve(int n) {
        if (n < 2) {
            return new ArrayList<>();
        }

        boolean[] isPrime = new boolean[n + 1];
        Arrays.fill(isPrime, true);
        isPrime[0] = isPrime[1] = false;

        List<Integer> primes = new ArrayList<>();

        for (int i = 2; i <= n; i++) {
            if (isPrime[i]) {
                primes.add(i);
            }

            for (int p : primes) {
                if (i * p > n) break;
                isPrime[i * p] = false;
                if (i % p == 0) break;

            }
        }
        return primes;
    }
```

Рис. 1: Eratosphen Linear Sieve