Решето Эратосфена

Древнегреческий математик Эратосфен предложил следующий алгоритм для нахождения всех простых, не превосходящих данного числа *n*. Возьмем массив S длины *n* и заполним его единицами (*пометим как невычеркнутые*). Теперь будем последовательно просматривать элементы S[*k*], начиная с *k* = 2. Если S[*k*] = 1, то заполним нулями (*вычеркнем* или *высеем*) все последующие ячейки, номера которых кратны *k*. В результате получим массив, в котором ячейки содержат 1 тогда и только тогда, когда номер ячейки — простое число.

Много времени можно сэкономить, если заметить, что, поскольку у составного числа, меньшего *n*, по крайней мере один из делителей не превосходит $k = \sqrt{n}$, процесс высевания достаточно закончить на $k = \sqrt{n}$.
Еще немного операций можно сэкономить, если — по той же причине — начинать вычеркивать кратные *k*, начиная не с *2k*, а с номера k^2.

Первое вычеркивание требует *n*/2 действий, второе — *n*/3, третье — *n*/5 и т. д. По формуле Мертенса

$$\sum_{\substack{p \le n \\ p \text{ простое}}} \frac{1}{p} \sim \log \log n,$$

так что для решета Эратосфена потребуется $O(n \log \log n)$ операций. Потребление памяти же составит $O(n)$.

Псевдокод алгоритма «Решето Эратосфена»

```
Вход: натуральное число n

Пусть A — булевый массив, индексируемый числами от 2 до n,
изначально заполненный значениями true.

 для i := 2, 3, 4, ..., пока i² ≤ n:
  если A[i] = true:
    для j := i², i² + i, i² + 2i, ..., пока j ≤ n:
      A[j] := false

Выход: числа i, для которых A[i] = true.
```