Обзор алгоритмов поиска и распознавания простых чисел, информация об их применимости.

Курицын Михаил
Люлькова Елена
Сизов Илья

Содержание

- Простое число
- Зачем искать простые числа?
- Алгоритмы поиска простых чисел
- Сравнение алгоритмов поиска простых чисел
- Алгоритмы распознавания простых чисел. Тесты простоты.
- Сравнение тестов простоты
- Список литературы

Простое число

- Простое число – это натуральное число, которое имеет ровно два различных натуральных делителя: единицу и самого себя.
- Остальные числа, кроме единицы, называются составными.
- Последовательность простых чисел начинается так: 2, 3, 5, 7, 11, 13, 17, 19, 23 , 29, 31, …

Самое большое простое число

- Один из рекордов поставил в своё время Эйлер, найдя простое число

$$2^{31} - 1 = 2147483647.$$

- Наибольшим известным простым числом по состоянию на февраль 2011 года является

$$2^{43112609} - 1$$

- За нахождение простых чисел из более чем 100 000 000 и 1 000 000 000 десятичных цифр EFF назначила денежные призы соответственно в 150 000 и 250 000 долларов.

Зачем искать простые числа?

- **Криптография** – наука о методах обеспечения конфиденциальности (невозможности прочтения информации посторонним) и аутентичности (целостности и подлинности авторства) информации.

- Криптография изучает методы шифрования информации – преобразования открытого текста на основе секретного алгоритма и/или ключа в шифрованный текст.

- В криптографических алгоритмах одной из важных задач является проверка на простоту, т.е. умение быстро отличить просто число от составного.

Алгоритмы поиска простых чисел

Простые способы нахождения начального списка простых чисел вплоть до некоторого значения дают :

- Решето Эратосфена
- Решето Сундарама
- Решето Аткина

Решето Эратосфена

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Простые числа:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47.

Алгоритм:

1. Пусть p = 2 (первому простому числу).
2. Считая от p, шагами по p, зачеркнуть в списке все числа от 2p до n.
3. Найти первое не зачеркнутое число, большее чем p, и присвоить значение переменной p это число.
4. Повторять шаги 3 и 4 до тех пор, пока p не станет больше чем n.

Решето Эратосфена

Сложность алгоритма:

$$O(n * log_2 log_2(n))$$

Решето Сундарама

1	2	3	~~4~~	5	6	~~7~~	8	9	~~10~~
11	~~12~~	~~13~~	14	15	~~16~~	~~17~~	18	~~19~~	20

$i = 1,$
$j = 1,\ldots,6;$

$i = 2,$
$j = 1,2,3;$

Простые числа:
3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41.

Алгоритм:

1. Из ряда натуральных чисел от 1 до N исключаются все числа вида $i + j + 2ij$ ($i = 1,2, \ldots, \frac{\sqrt{2N+1}-1}{2}$; $j = i, i+1, \ldots, \frac{N-i}{2i+1}$).

2. Каждое из оставшихся чисел умножается на 2 и увеличивается на 1. Полученная в результате последовательность представляет собой все нечётные простые числа в отрезке $[1, 2N+1]$.

Решето Сундарама. Обоснование

- Алгоритм работает с нечётными натуральными числами большими 1, представленными в виде $2m+1$, где m является натуральным числом.
- Если число $2m+1$ является составным, то оно представляется в виде произведения двух нечётных чисел больших единицы, то есть:

$2m+1 = (2i+1)(2j+1)$, где i, j – натуральные числа

Что эквивалентно: $m = 2ij+i+j$

- Если из ряда натуральных чисел исключить все числа вида $2ij + i + j$, , то для каждого из оставшихся чисел m число $2m+1$ обязано быть простым.

- Если число $2m+1$ является простым, то число m невозможно представить в виде $2ij+i+j$ и, таким образом, m не будет исключено в процессе работы алгоритма.

Решето Аткина

В основу алгоритма "решета Аткина" положены три стандартные теоремы теории элементарных чисел:

1. *n – простое, если:*
$4 * x^2 + y^2 = n \ (x > 0, y > 0)$
n mod 4 = 1
n – нечетное число

2. *n – простое, если:*
$3 * x^2 + y^2 = n \ (x > 0, y > 0)$
n mod 6 = 1
n – нечетное число

3. *n – простое, если:*
$3 * x^2 - y^2 = n \ (x > 0, y > 0)$
n mod 12 = 11
n – нечетное число

Алгоритм

- Создать **решето** (массив соответствия простым числам для всех положительных, целых чисел начиная с 2). Изначально все элементы решета помечаются как составные.

- Для каждого числа n в решете , если остаток от деления по модулю 60:
  - Равен 1, 13, 17, 29, 37, 41, 49, или 53, и $n = 4 * x^2 + y^2$ поменять значение в решете на противоположное.
  - Равен 7, 19, 31, или 43, и $n = 3 * x^2 + y^2$; поменять значение решете на противоположное.
  - Равен 11, 23, 47, или 59, и $n = 3 * x^2 - y^2$ при($x > y$); поменять значение в решете на противоположное.

  (x и y целые, положительные числа)

- Взять наименьшее число из решета, помеченное как простое, и пометить все элементы решета, кратные квадрату этого простого числа как составные.

- Повторить шаг 3

Решето Аткина

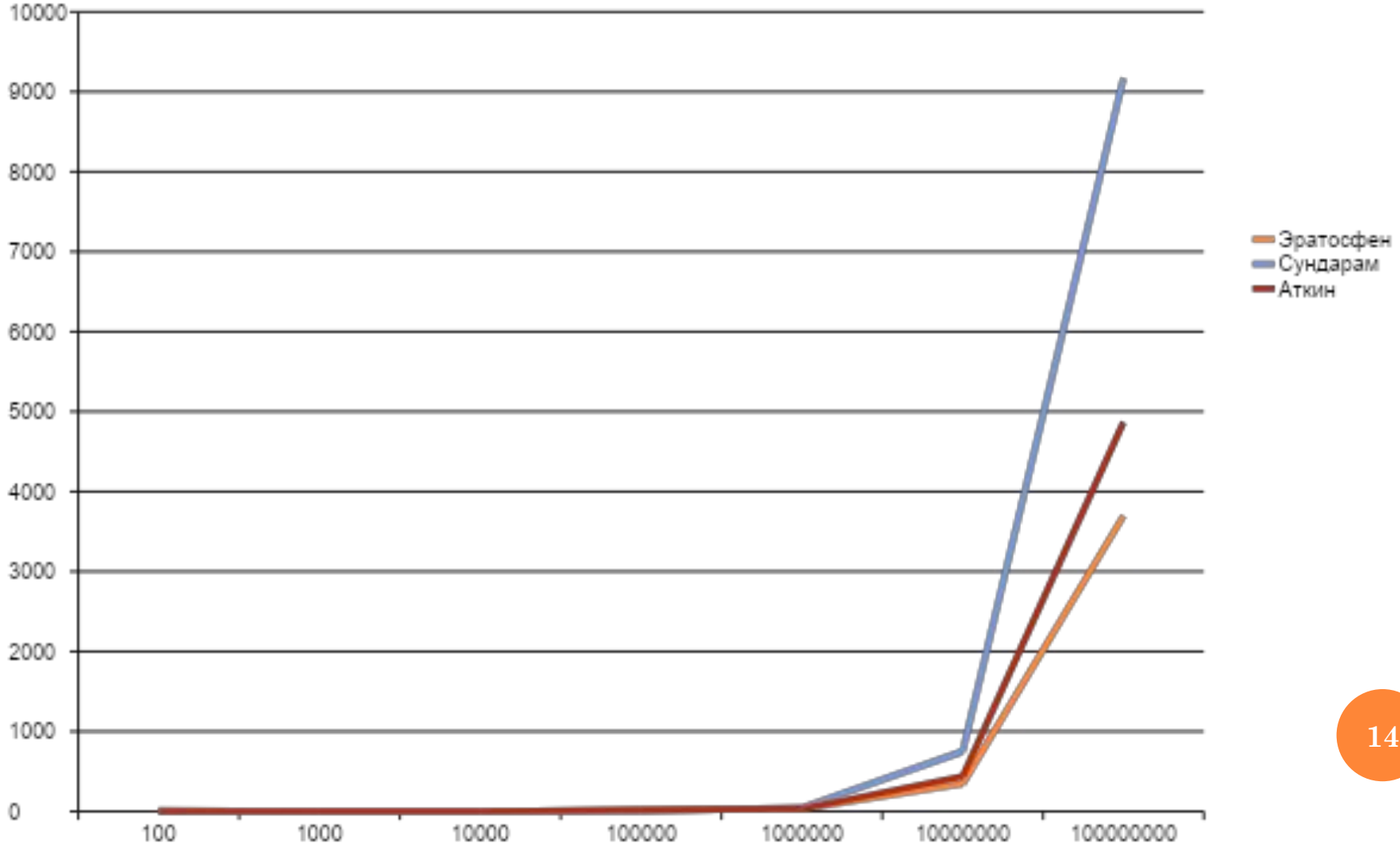
Алгоритм имеет асимптотическую сложность:

$$O\left(\frac{N}{\log\log N}\right)$$

и требует следующее кол-во бит памяти:

$$O\left(N^{\frac{1}{2}+o(1)}\right)$$

Сравнение алгоритмов поиска простых чисел



Алгоритмы распознавания простых чисел. Тесты простоты

Тест простоты — алгоритм, который по заданному натуральному числу определяет, простое ли это число.

- Перебор делителей
- Теорема Вильсона
- Тест Ферма
- Тест Пепина
- Тест Миллера – Рабина
- Тест Агравала – Каяла – Саксены

Перебор делителей

Перебор делителей — алгоритм тестирования простоты числа путем полного перебора всех возможных потенциальных делителей.

Алгоритм:

1. Перебор всех целых чисел от 2 до квадратного корня из числа n и вычисление остатка от деления n на каждое из этих чисел.

2. Если остаток от деления на некоторое число m равен нулю, то m является делителем n. В этом случае либо n объявляется составным, и алгоритм заканчивает работу.

3. По достижении квадратного корня из n и невозможности сократить n ни на одно из меньших чисел, n объявляется простым.

Теорема Вильсона

Теорема Вильсона — теорема теории чисел, которая утверждает, что

p — простое число тогда и только тогда, когда $(p - 1)! + 1$ делится на p

Тест Ферма

Основан на теореме Ферма, которая гласит:

Если p – простое число, то для любого целого a выполняется равенство

$$a^{p-1} \equiv 1 (mod\ p)$$

или

$(a^{p-1} - 1)$ делится на p нацело.

Примечание:

Для составных p истинность равенства маловероятна.

Тест Пепина

- Тест пепина является тестом простоты для чисел Ферма. Число ферма – это число вида: $F_n = 2^{2^n} + 1$, n – целое, неотрицательное.

- **Число Ферма является простым тогда и только тогда, когда $3^{(F_n-1)/2} \equiv -1 \ (mod\ F_n)$.**

- На сегодняшний день известно только 5 простых чисел Ферма: 3, 5, 17, 257 и 65537.

Тест Миллера - Рабина

- **Тест Миллера – Рабина** - вероятностный полиномиальный тест простоты.
- Тест позволяет эффективно определять, является ли данное число составным. Однако, с его помощью нельзя строго доказать простоту числа.

Свидетели простоты и теорема Рабина

Пусть m – нечетное число большее 1. Тогда m-1 представимо в виде:

$$m\text{-}1 = 2^s * t \text{ , где } t - \text{нечетно}$$

Целое число a, 1<a<m, называется свидетелем простоты m, если выполняется одно из условий:

$$a^t \bmod m = 1$$

или

существует такое r, $(a^t)^{2^r} \bmod m = -1$

Тест Миллера - Рабина

Алгоритм:

Параметром алгоритма Миллера – Рабина является количество раундов r. В каждом раунде выполняются следующие действия:

1. Выбирается случайное число a, $2 < a < m-1$.

2. Если a не является свидетелем простоты числа m, то выдается ответ *«m составное»*, и алгоритм завершается. Иначе, выбирается новое случайное число a и процедура проверки повторяется.

3. После нахождения r свидетелей простоты, выдается ответ *«m, вероятно, простое»*, и алгоритм завершается.

Тест Миллера - Рабина

Сложность алгоритма :

$$O\ (log^3 n)$$

Однако, правильность работы алгоритма не всегда является доказанной. Вероятность, что составное число не будет выявлено за время t, обычно не превосходит

$$e^{-\alpha t}$$

Тест Агравала — Каяла — Саксены (или тест AKS)

- *Универсальность*: Тест AKS может использоваться для проверки простоты любых чисел.
- *Полиномиальность*: Наибольшее время работы алгоритма ограничено полиномом от количества цифр в проверяемом числе.
- *Детерминизм*: Алгоритм гарантирует получение ответа.
- *Безусловность*: Корректность теста AKS не зависит от каких-либо недоказанных гипотез.

Тест Агравала — Каяла — Саксены (или тест AKS)

Основные идеи и принципы, на котором основан алгоритм AKS:

Утверждение:

n – простое тогда и только тогда, когда:

$$НОД\ (a, n) = 1$$

$$(x - a)^n \equiv (x^n - a)(mod\ n)$$

Теорема AKS

Пусть $n \geq 2; n$ – целое; q, r – простые числа , причем

1. $\forall m \in \{1,2,..,r\}: НОД(m, n) = 1$
2. q – наибольший простой делитель *(r-1)*
3. $q \geq 4 * \sqrt{r} log_2 n$
4. $n^{(r-1)/q}! \equiv 1(mod\ r)$
5. $\forall a\{1,2,\dots,2 * \sqrt{r log_2 n} + 1\} : (x - a)^n \equiv (x^n - a)(mod\ n, x^r - 1)$

Тогда n – степень простого числа.

Тест Агравала — Каяла — Саксены (или тест AKS)

Алгоритм:

$r = 2$
$while(r < n)\{$
$if(\text{НОД}(r, n) \neq 1)$ вернуть составное;
$if(r -$ простое $, r > 2)\{$
$q -$ наибольший простой делитель у $(r - 1)$;

$if\ (q \geq 4\sqrt{r}log_2 n)\ and\ (n^{(r-1)/q} \neq 1(mod\ r)))\ break;$
$\}$
$r = r + 1;$
$\}$
$for\ a = 1\ to\ (\lfloor 2\sqrt{r}log_2 n \rfloor + 1)$
$if\ \left((x - a)^n \equiv (x^n - a)(mod\ x^r - 1, n)\right)$ вернуть составное;
$if\ \left(n = a^b; a, b -\right.$ целые; $\left.a, b \geq 2\right)$ вернуть составное;
вернуть простое;

Тест Агравала — Каяла — Саксены (или тест AKS)

Примечание:

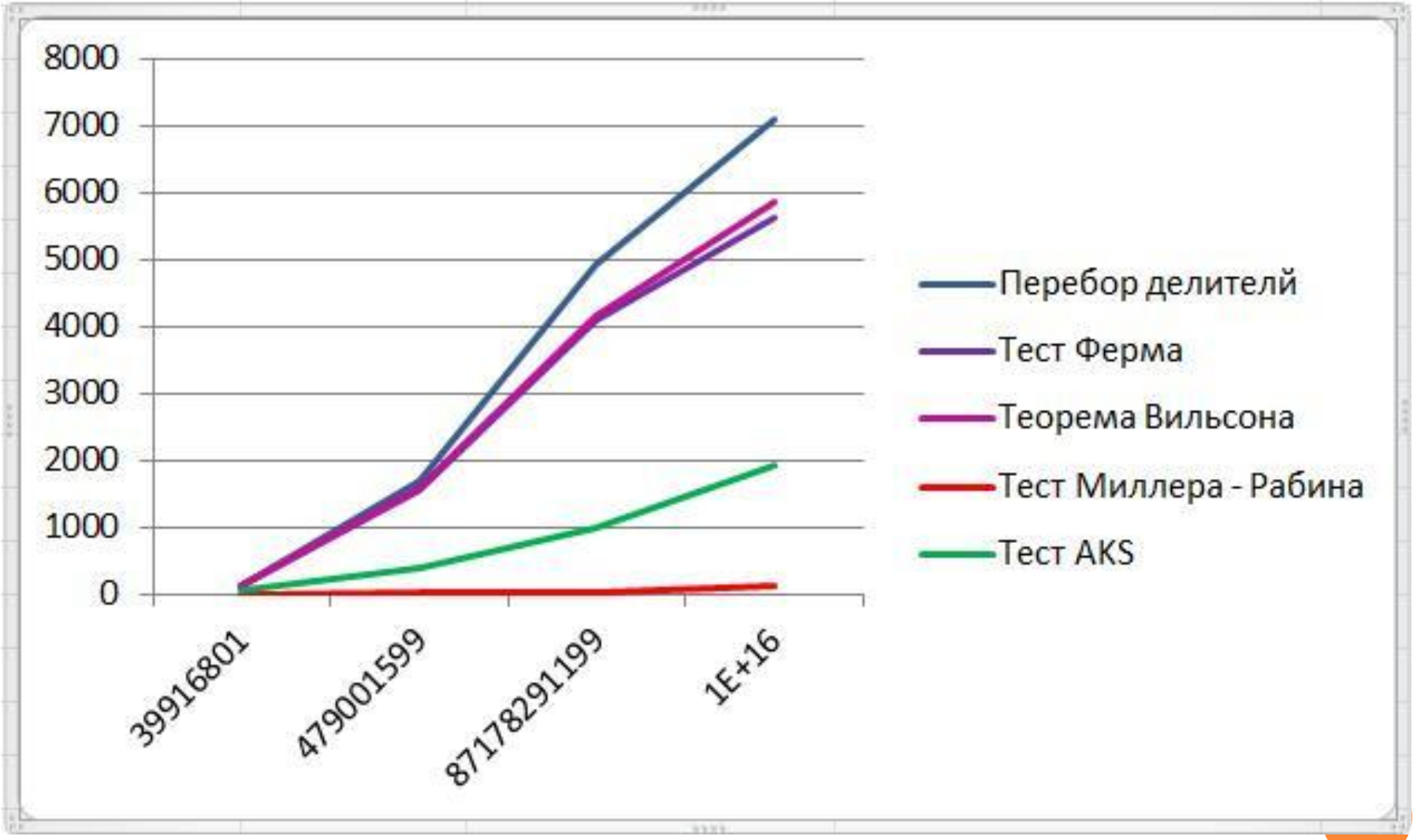
Выражение: $\left((x-a)^n \equiv (x^n - a)(mod\ x^r - 1, n)\right)$ означает следующее:
для многочленов $(x-a)^n$ и $(x^n - a)$ найдется многочлен $q(x) \epsilon Z[x]$ (кольцо многочленов от x с целыми коэффициентами) такой, что все коэффициенты многочлена

$$(x-a)^n - (x^n - a) - (x^r - 1) * q(x) \text{ кратны } n.$$

Сложность алгоритма AKS:

$$O\ (log^{19} n)$$

Сравнение тестов простоты



Список литературы

- *Википедия*
- *Л. Бараш*, Алгоритм AKS проверки чисел на простоту и поиск констант генераторов псевдослучайных чисел.
- *С.В. Сизый*, Лекции по теории чисел.
- *С. Г. Гиндикин*, Малая теорема Ферма / Квант. — 1972. — № 10.
- *A.O.L. Atkin, D.J. Bernstein*, Prime sieves using binary quadratic forms. – 1999.
- *И.В.Агафонова*, Проверка чисел на простоту: полиномиальный алгоритм.
- *Б.А. Фороузан*, Математика криптографии и теория шифрования.

Спасибо за внимание!