Create account Log in

Article Talk Read Edit View history Search

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
العربية
Deutsch
Español
فارسی
Français
Italiano
Bahasa Melayu
Polski
Português
Русский
Tiếng Việt
Edit links

Sieve of Atkin

From Wikipedia, the free encyclopedia

In mathematics, the **sieve of Atkin** is a fast, modern algorithm for finding all prime numbers up to a specified integer. It is an optimized version of the ancient sieve of Eratosthenes which does some preliminary work and then marks off multiples of the square of each prime, rather than multiples of the prime itself. It was created in 2003 by A. O. L. Atkin and Daniel J. Bernstein.[1]

Contents [hide]

Algorithm [edit]

In the algorithm:

- All remainders are modulo-sixty remainders (divide the number by sixty and return the remainder).
- All numbers, including x and y, are positive integers.
- Flipping an entry in the sieve list means to change the marking (prime or nonprime) to the opposite marking.
- This results in numbers with an odd number of solutions to the corresponding equation being potentially prime (prime if they are also square free), and numbers with an even number of solutions being composite.

The algorithm:

1. Create a results list, filled with 2, 3, and 5.
2. Create a sieve list with an entry for each positive integer; all entries of this list should initially be marked non prime (composite).
3. For each entry number n in the sieve list, with modulo-sixty remainder r :
    1. If r is 1, 13, 17, 29, 37, 41, 49, or 53, flip the entry for each possible solution to $4x^2 + y^2 = n$. The number of flipping operations as a ratio to the sieving range for this step approaches $\frac{4\sqrt{\pi}}{15}$[1] × $\frac{8}{60}$ (the "8" in the fraction comes from the eight modulos handled by this quadratic and the 60 because Atkin calculated this based on an even number of modulo 60 wheels), which results in a fraction of about 0.1117010721276....
    2. If r is 7, 19, 31, or 43, flip the entry for each possible solution to $3x^2 + y^2 = n$. The number of flipping operations as a ratio to the sieving range for this step approaches $\pi\sqrt{0.12}$[1] × $\frac{4}{60}$ (the "4" in the fraction comes from the four modulos handled by this quadratic and the 60 because Atkin calculated this based on an even number of modulo 60 wheels), which results in a fraction of about 0.072551974569....
    3. If r is 11, 23, 47, or 59, flip the entry for each possible solution to $3x^2 - y^2 = n$ when $x > y$. The number of flipping operations as a ratio to the sieving range for this step approaches $\sqrt{1.92 ln(\sqrt{0.5}+\sqrt{1.5})}$[1] × $\frac{4}{60}$ (the "4" in the fraction comes from the four modulos handled by this quadratic and the 60 because Atkin calculated this based on an even number of modulo 60 wheels), which results in a fraction of about 0.060827679704....
4. If r is something else, ignore it completely.
5. Start with the lowest number in the sieve list.
6. Take the next number in the sieve list still marked prime.
7. Include the number in the results list.
8. Square the number and mark all multiples of that square as non prime. Note that the multiples that can be factored by 2, 3, or 5 need not be marked, as these will be ignored in the final enumeration of primes.
9. Repeat steps five through eight. The total number of operations for these repetitions of marking the

squares of primes as a ratio of the sieving range is the sum of the inverse of the primes squared, which approaches the prime zeta function(2) of 0.45224752004... minus $\frac{1}{2^2}$, $\frac{1}{3^2}$, and $\frac{1}{5^2}$ for those primes which have been eliminated by the wheel, with the result multiplied by $\frac{16}{60}$ for the ratio of wheel hits per range; this results in a ratio of about 0.01363637571....

Adding the above ratios of operations together, the above algorithm takes a constant ratio of flipping/marking operations to the sieving range of about 0.2587171021...; From an actual implementation of the algorithm, the ratio is about 0.25 for sieving ranges as low as 67.

Pseudocode [edit]

The following is pseudocode which **combines** Atkin's algorithms 3.1, 3.2, and 3.3[1] by using a **combined** set "s" of all the numbers modulo 60 excluding those which are factors of the prime numbers 2, 3, and 5, as per the algorithms, for a straightforward version of the algorithm that supports optional bit packing of the wheel; although not specifically mentioned in the referenced paper, this pseudocode eliminates some obvious combinations of odd/even x's/y's in order to reduce computation where those computations would never pass the modulo tests anyway (i.e. would produce even numbers or modulo's other than those tested):

```
limit ← 1000000000        // arbitrary search limit

// set of wheel "hit" positions for a 2/3/5 wheel rolled twice as per the Atkin
algorithm
s ← {1,7,11,13,17,19,23,29,  31,37,41,43,47,49,53,59}

// Initialize the sieve with enough wheels to include limit:
for n ← 60 × w + x where w ∈ {0,1,...,limit ÷ 60}, x ∈ s:
    is_prime(n) ← false

// Put in candidate primes:
//   integers which have an odd number of
//   representations by certain quadratic forms.
// Algorithm step 3.1:
for n ≤ limit, n ← 4x²+y² where x ∈ {1,2,...} and y ∈ {1,3,...} // all x's odd y's
    if n mod 60 ∈ {1,13,17,29,37,41,49,53}:
        is_prime(n) ← ¬is_prime(n)   // toggle state
// Algorithm step 3.2:
for n ≤ limit, n ← 3x²+y² where x ∈ {1,3,...} and y ∈ {2,4,...} // only odd x's
    if n mod 60 ∈ {7,19,31,43}:                                 // and even y's
        is_prime(n) ← ¬is_prime(n)   // toggle state
// Algorithm step 3.3:
for n ≤ limit, n ← 3x²-y² where x ∈ {2,3,...} and y ∈ {x-1,x-3,...,1} //all
even/odd
    if n mod 60 ∈ {11,23,47,59}:                                   // odd/even
combos
        is_prime(n) ← ¬is_prime(n)   // toggle state

// Eliminate composites by sieving, only for those occurrences on the wheel:
for n² ≤ limit, n ← 60 × w + x where w ∈ {0,1,...}, x ∈ s, n ≥ 7:
    if is_prime(n):
        // n is prime, omit multiples of its square; this is sufficient
        // because square-free composites can't get on this list
        for c ≤ limit, c ← n² × (60 × w + x) where w ∈ {0,1,...}, x ∈ s:
            is_prime(c) ← false

// one sweep to produce a sequential list of primes up to limit:
output 2, 3, 5
for 7 ≤ n ≤ limit, n ← 60 × w + x where w ∈ {0,1,...}, x ∈ s:
    if is_prime(n): output n
```

This pseudocode is written for clarity; although some redundant computations have been eliminated by controlling the odd/even x/y combinations, it still wastes almost half of its quadratic computations on non-productive loops that don't pass the modulo tests such that it will not be faster than an equivalent wheel factorized (2/3/5) sieve of Eratosthenes. To improve its efficiency, a method must be devised to minimize or eliminate these non-productive computations.

Explanation [edit]

The algorithm completely ignores any numbers with remainder modulo 60 that is divisible by two, three, or five, since numbers with a modulo 60 remainder divisible by one of these three primes are themselves divisible by that prime.

All numbers n with modulo-sixty remainder 1, 13, 17, 29, 37, 41, 49, or 53 have a modulo-four remainder of 1. These numbers are prime if and only if the number of solutions to $4x^2 + y^2 = n$ is odd and the number is squarefree (proven as theorem 6.1 of[1]).

All numbers n with modulo-sixty remainder 7, 19, 31, or 43 have a modulo-six remainder of 1. These numbers are prime if and only if the number of solutions to $3x^2 + y^2 = n$ is odd and the number is squarefree (proven as theorem 6.2 of[1]).

All numbers n with modulo-sixty remainder 11, 23, 47, or 59 have a modulo-twelve remainder of 11. These numbers are prime if and only if the number of solutions to $3x^2 - y^2 = n$ is odd and the number is squarefree (proven as theorem 6.3 of[1]).

None of the potential primes are divisible by 2, 3, or 5, so they can't be divisible by their squares. This is why squarefree checks don't include 2^2, 3^2, and 5^2.

Computational complexity [edit]

It can be computed[1] that the above series of three quadratic equation operations each have a number of operations that is a constant ratio of the range as the range goes to infinity; as well the prime square free culling operations can be described by the prime zeta function(2) with constant offsets and factors so it is also a constant factor of the range as the range goes to infinity. Therefore, the algorithm described above can compute primes up to N using $O(N)$ operations with only $O(N)$ bits of memory.

The page segmented version implemented by the authors has the same $O(N)$ operations but reduces the memory requirement to just that required by the base primes below the square root of the range of $O(N^{1/2}/\log N)$ bits of memory plus a minimal page buffer. This is slightly better performance with the same memory requirement as the page segmented sieve of Eratosthenes which uses $O(N \log \log N)$ operations and the same $O(N^{1/2}/\log N)$ bits of memory[2] plus a minimal page buffer. However, such a sieve does not outperform a Sieve of Eratosthenes with maximum practical wheel factorization (a combination of a 2/3/5/7 sieving wheel and pre-culling composites in the segment page buffers using a 2/3/5/7/11/13/17/19 pattern), which although it has slightly more operations than the Sieve of Atkin for very large but practical ranges, has a constant factor of less complexity per operation by about three times in comparing the per operation time between the algorithms implemented by Bernstein in CPU clock cycles per operation. The main problem with the Page Segmented Sieve of Atkin is the difficulty in implementing the "prime square free" culling sequences due to the span between culls rapidly growing far beyond the page buffer span; the time expended for this operation in Bernstein's implementation rapidly grows to many times the time expended in the actual quadratic equation calculations, meaning that the linear complexity of the part that would otherwise be quite negligible becomes a major consumer of execution time. Thus, even though an optimized implementation may again settle to a O(n) time complexity, this constant factor of increased time per operations means that the Sieve of Atkin is slower.

A special modified "enumerating lattice points" variation **which is not the above version** of the Sieve of Atkin can theoretically compute primes up to N using $O(N/\log \log N)$ operations with $N^{1/2 + o(1)}$ bits of memory[1] but this variation is rarely if ever implemented, including by the authors. That is a little better in performance at a very high cost in memory as compared to both the ordinary page segmented version and to an equivalent but rarely if ever implemented version of the sieve of Eratosthenes which uses $O(N)$ operations and $O(N^{1/2}(\log \log N)/\log N)$ bits of memory.[3][4][5]

Pritchard observed that for the Wheel Sieves, one can reduce memory consumption while preserving Big O time complexity, but this generally comes at a cost in a increased constant factor for time per operation due to the extra computational complexities; one would assume that this is also true for the special variation of the Sieve of Atkin as per the above discussion. Therefore, this special version is likely more of value as an intellectual exercise than a practical prime sieve with reduced real time expended for a given large practical sieving range.

See also [edit]

- Sieve of Sundaram
- Sieve theory

References [edit]

1. ^ a b c d e f g h i j A.O.L. Atkin, D.J. Bernstein, *Prime sieves using binary quadratic forms*, Math. Comp. **73** (2004), 1023-1030.[1]

2. ^ Pritchard, Paul, "Linear prime-number sieves: a family tree," *Sci. Comput. Programming* **9**:1 (1987), pp. 17–35.
3. ^ Paul Pritchard, A sublinear additive sieve for finding prime numbers, Communications of the ACM 24 (1981), 18–23. MR 82c:10011
4. ^ Paul Pritchard, Explaining the wheel sieve, Acta Informatica 17 (1982), 477–485. MR 84g:10015
5. ^ Paul Pritchard, Fast compact prime number sieves (among others), Journal of Algorithms 4 (1983), 332–344. MR 85h:11080

External links [edit]

- An optimized implementation of the sieve (in C)

v · t · e	Number-theoretic algorithms [hide]
Primality tests	AKS test · APR test · Baillie–PSW · ECPP test · Elliptic curve · Pocklington · Fermat · Lucas · *LUCAS–LEHMER* · *LUCAS–LEHMER–RIESEL* · *PROTH'S THEOREM* · *PÉPIN'S* · Quadratic Frobenius test · Solovay–Strassen · Miller–Rabin
Prime-generating	**Sieve of Atkin** · Sieve of Eratosthenes · Sieve of Sundaram · Wheel factorization
Integer factorization	Continued fraction (CFRAC) · Dixon's · Lenstra elliptic curve (ECM) · Euler's · Pollard's rho · *p* − 1 · *p* + 1 · Quadratic sieve (QS) · General number field sieve (GNFS) · *Special number field sieve (SNFS)* · Rational sieve · Fermat's · Shanks' square forms · Trial division · Shor's
Multiplication	Ancient Egyptian · Long · Karatsuba · Toom–Cook · Schönhage–Strassen · Fürer's
Discrete logarithm	Baby-step giant-step · Pollard rho · Pollard kangaroo · Pohlig–Hellman · Index calculus · Function field sieve
Greatest common divisor	Binary · Euclidean · Extended Euclidean · Lehmer's
Modular square root	Cipolla · Pocklington's · Tonelli–Shanks
Other algorithms	Chakravala · Cornacchia · Integer relation · Integer square root · Modular exponentiation · Schoof's
Italics indicate that algorithm is for numbers of special forms · Smallcaps indicate a deterministic algorithm	

Categories: Primality tests