NOTES ON PRIME NUMBER SIEVES

ERIC MARTIN

1. Eratosthenes' sieve

Let $N \geq 2$ be given.

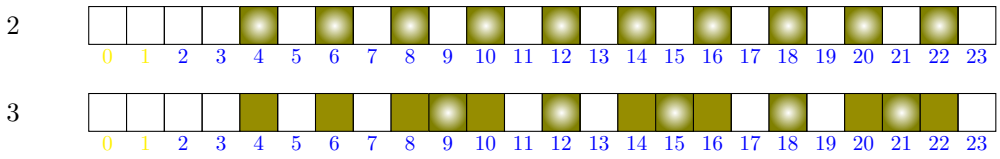
In the first version of the sieve, we use a list meant to denote 0, 1, 2, 3, 4,... N, and we generate all prime numbers at most equal to N by flagging to False:

- all positive multiples of $p_1 = 2$ at most equal to N, 2 excepted;
- all positive multiples of $p_2 = 3$ at most equal to N, 3 excepted;
- ...
- all multiples of p_k at most equal to N, p_k excepted, where p_k is the least number greater than $p_1, \ldots, p_{k-1}$ not flagged to False so far.
- ...

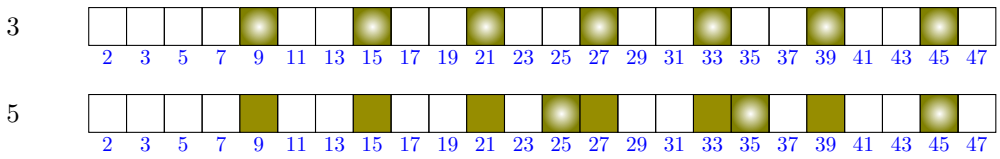
For all $k \geq 1$, p_k is the kth prime number.

We stop when p_k exceeds $\sqrt{N}$ since the smallest prime factor of a nonprime number n is at least equal to $\sqrt{n}$.

At stage k, $p_k \times 2, \ldots, p_k \times p_{k-1}$ have been flagged to False when we did so with the multiples of $p_1, \ldots, p_{k-1}$ at stage $1, \ldots, k-1$, respectively. Hence at stage k, we flag to False the multiples of p_k starting from p_k^2.



In the second version of the sieve, we use a list L meant to denote 2, 3, 5, 7, 9,..., N or 2, 3, 5, 7, 9,..., $N-1$ depending on whether N is odd or even, respectively, which saves half the space and avoids having to flag to False the multiples of 2. Let n be an odd number. The index k of the representative of n in L is $\frac{n-1}{2}$, hence the index of the representative of n^2 is $\frac{n^2-1}{2} = \frac{(n-1)(n+1)}{2} = \frac{n-1}{2}2(\frac{n-1}{2}+1) = 2k(k+1)$. Also, we only flag to False the odd multiples of n: $3n$, $5n$, $7n$.... So after having flagged to False a number a, we flag to False $a+2n$, whose representative has index $\frac{a+2n-1}{2} = \frac{a-1}{2} + n = \frac{a-1}{2} + 2k + 1$, so we add $2k+1$ to the index of the representative of a.



Date: Session 1, 2016.

2. Euler's sieve

Note that Eratosthenes' sieve can flag to False a composite number more than once. Euler's sieve avoids this by removing a number from the list rather than flagging it. We start with the list $[2, 3, 4, 5, \dots, N]$, and we generate all prime numbers at most equal to N by removing from the list:

- using 2, the first number in the list:
  - 2^2, 2^3, 2^4,..., up to 2^r for the largest r with $2^r \leq N$,
  - 2×3, $2^2 \times 3$, $2^3 \times 3$,..., up to $2^r \times 3$ for the largest r with $2^r \times 3 \leq N$,
  - as 4 is no longer in the list, 2×5, $2^2 \times 5$, $2^3 \times 5$,..., up to $2^r \times 5$ for the largest r with $2^r \times 5 \leq N$,
  - as 6 is no longer in the list, 2×7, $2^2 \times 7$, $2^3 \times 7$,..., up to $2^r \times 7$ for the largest r with $2^r \times 7 \leq N$,
  - as 8 is no longer in the list, 2×9, $2^2 \times 9$, $2^3 \times 9$,..., up to $2^r \times 9$ for the largest r with $2^r \times 9 \leq N$,
  - ...
- using 3, the next number in what remains of the list:
  - 3^2, 3^3, 3^4,..., up to 3^r for the largest r with $3^r \leq N$,
  - as 4 is no longer in the list, 3×5, $3^2 \times 5$, $3^3 \times 5$,..., up to $3^r \times 5$ for the largest r with $3^r \times 5 \leq N$,
  - as 6 is no longer in the list, 3×7, $3^2 \times 7$, $3^3 \times 7$,..., up to $3^r \times 7$ for the largest r with $3^r \times 7 \leq N$,
  - as 8, 9 and 10 are no longer in the list, 3×11, $3^2 \times 11$, $3^3 \times 11$,..., up to $3^r \times 11$ for the largest r with $3^r \times 11 \leq N$,
  - ...
- ...

We also stop when the next number in what remains in the list exceeds $\sqrt{N}$.

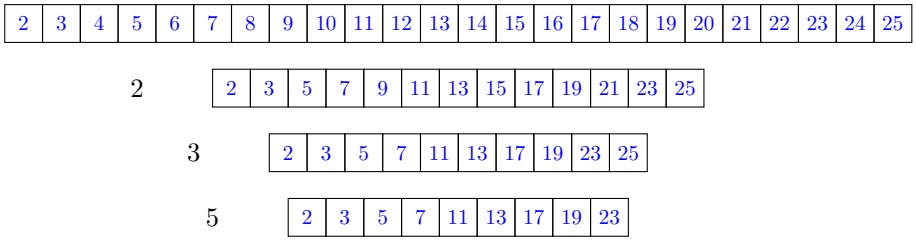
Note that if using 2, the first number in the list, we removed from the list:

- 2×2,
- 2×3,
- as 4 is no longer in the list, 2×5,
- ...

then 2^3 would incorrectly remain in the list.

At stage k, all strictly positive multiples of the kth prime number p_k at most equal to N, with the exception of p_k itself, are removed from the list. This is verified by induction. Indeed, if during stage k, a number n in $\{2, 3, \dots, N\}$ with $p_k n \leq N$ is not considered then:

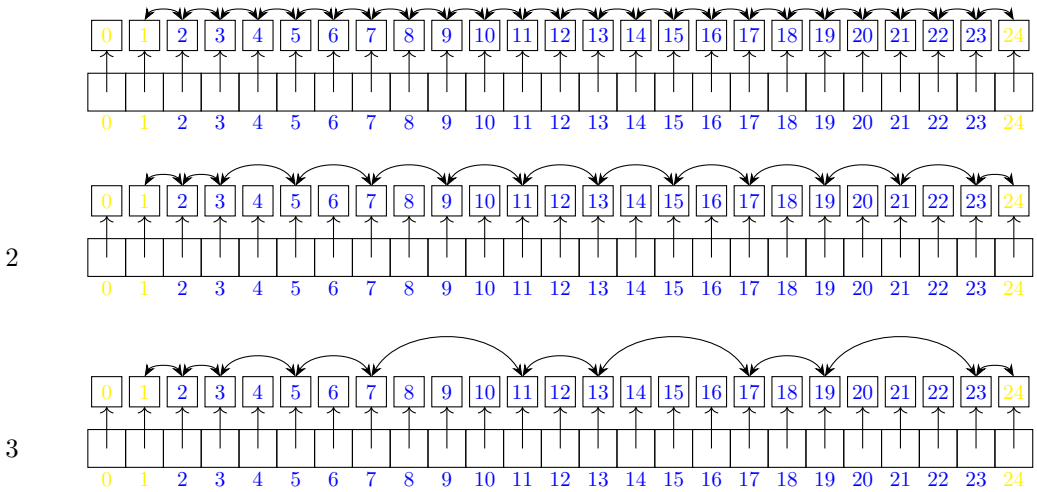
- either n is smaller than p_k, in which case it is a multiple of at least one of p_1, ..., p_{k-1}, which implies that $p_k \times n$ is also a multiple of at least one of p_1, ..., p_{k-1}, so by inductive hypothesis, $p_k \times n$ was removed from the list during one of the previous stages,
- or n is greater than p_k but no longer belongs to the list (it is a number such as 4, 6, 8 at stage 1, or a number such as 4, 6, 8, 9, 10 at stage 2), in which case:
  - either n was removed during one of the previous stages, hence n is a multiple of at least one of p_1, ..., p_{k-1}, which implies as in the previous case that $p_k \times n$ was also removed from the list,
  - or n is a multiple of p_k which was removed earlier in the current stage, so n is a number of the form $p_k^r m$ for some $r \geq 1$ and some number m which was then found in what remained of the list, therefore $p_k n = p_k^{r+1} m$ was also removed from the list earlier in the current stage.



Removing an element from an ordered list, keeping the remaining elements in order, is on average linear in the length of the list. In the second version of the sieve, at every stage k, we create a set consisting of the numbers in what remains of the list, and remove from the set all products at most equal to N of p_k with a number at least equal to p_k in what remains of the list, that is, all multiples at most equal to N of p_k that have not been removed at earlier stages as multiples of numbers smaller than p_k. The cost of every such removal is constant. At the end of each stage, we sort what remains of

the set to get an updated version of what remains of the list, an operation which is of complexity $n\log(n)$ in the number of elements in the set. This second version is more efficient than the first one, but still less efficient than the first version of Eratosthenes' sieve.

In the third version of the sieve, we use a list $[\text{Node}(0), \text{Node}(1), \text{Node}(2), \text{Node}(3), \ldots, \text{Node}(N+1)]$ where for all $n \leq N+1$, Node(n) is an object consisting of the value n, a link "next" to $N(n+1)$ (None if $n = N+1$), and a link "previous" to $N(n-1)$ (None if $n = 0$). Removing a number n is achieved by letting Node(n).next.previous change from Node(n) to Node(n).previous, and Node(n).previous.next change from Node(n) to Node(n).next; since n is accessed using n itself as index of the list, the cost of removing n is constant. Accessing the number after a number n in what remains of the list is done using Node(n).next, also an operation with a constant cost. This third version is more efficient than the second one, but still less efficient than the first version of Eratosthenes' sieve.



In the pictures above, we depict a link from A to B iff there is also a link from B to A, but the original link from 4 to 3, the original link from 4 to 5, ..., are not removed, and then the newly created link from 9 to 7, the newly created link from 9 to 11, ..., are not removed.

3. Sundaram's sieve

We use a list L meant to denote 1, 3, 5, 7, 9,..., N or 1, 3, 5, 7, 9,..., $N-1$ depending on whether N is even or odd, respectively. We flag to False all odd numbers greater than 1 and at most equal to N whose representatives have indexes of the form $i+j+2ij$ with $1 \leq i \leq j$. Correctness is shown as follows. Let an odd number n greater than 1 and at most equal to N be given. Write $n = 2k+1$ (so k is the index of the representative of n in L). Then n is flagged to False iff k is of the form $i+j+2ij$, that is, iff n is equal to $2(i+j+2ij)+1$, which is equal to $(2i+1)(2j+1)$. We conclude by observing that an odd number is prime iff it is not the product of two odd numbers greater than 1.

Note that if h is 1 plus the index of the last element of L, then when generating pairs (i,j), j is a valid index iff $i+j+2ij < h$, iff $j < \frac{h-i}{1+2i}$, iff $j < \lceil \frac{h-i}{1+2i} \rceil$.