

IAM 550, Fall 2019
Self test 02, 11/01/2019

Sieve of Eratosthenes: find prime numbers up to N

(https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes)

- Fill a vector A with N integers from 1 to N.
- Loop from 2 to N, index i. If A(i) is zero, next.
- Loop from i to N with index j. If $k=i*j$ is larger than N end this loop. Else, k is not a prime, so set A(k) to zero.
- After all loops have finished, all elements of A that are not zero are prime numbers. Print them out.

Note: this can be programmed more efficiently, but works just fine if N is not too large. For really large N this is not a good algorithm anyways because it needs too much memory. We leave it to the mathematicians to come up with better ones. Very large prime numbers are needed for public key encryption, see

https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange