

Number theory tasks for which efficient algorithms are known

Paul Dumais, October 2002

1 See textbook algorithm 5.1, *euclidean algorithm*.

Input: $m \geq 1, n \geq 1$

Output: $\gcd(m, n)$

2 See textbook algorithm 5.2, *extended euclidean algorithm*.

Input: $n \geq 2, x \in \mathbb{Z}_n^*$

Output: $x^{-1} \bmod n$

3 See textbook algorithm 5.5, *square-and-multiply*.

Input: $n \geq 1, x \geq 1, a \geq 1$

Output: $x^a \bmod n$

4 See textbook algorithm 5.7, *Miller-Rabin*.

Input: $n \geq 2$

Output: “ n is probably prime” or “ n is composite”

5 See textbook theorem 1.2.

Input: $n \geq 2$, the prime factorization of n

Output: $\phi(n)$

6 See textbook theorem 5.8, and *Handbook of Applied Cryptography* algorithm 4.80.

Input: $p \in \mathbb{P}$, the prime factorization of $p - 1$

Output: a generator of \mathbb{Z}_p^*

7 See textbook example 5.8, and *Handbook of Applied Cryptography* algorithm 2.149.

Input: an odd integer $n \geq 3, a \in \mathbb{Z}_n$

Output: $\left(\frac{a}{n}\right)$

8 See textbook section 5.5, and *Handbook of Applied Cryptography* algorithms 3.34, 3.36 and 3.44.

Input: $n \geq 2$, the prime factorization of $n, x \in \mathbb{Q}_n$

Output: $z \in \mathbb{Z}_n^*$ such that $z^2 \bmod n = x$