Review of Time Estimates for Algorithms

**Basic operations:** Addition, multiplication and division of \( m, n \) take \( O(\log(m) \log(n)) \) bit operations.

**Euclidean algorithm:** \( \gcd(m, n) : O(\log(m) \log(n)) \)

**Arithmetic mod \( m \):** Addition, multiplication and inverses/division take \( O(\log^2(m)) \).

**Chinese Remainder Theorem:** to solve \( x \equiv a \pmod{m} \), \( x \equiv b \pmod{n} \) takes \( O(\log(m) \log(n)) \).

**Fast exponentiation (power-mod):** computing \( a^n \pmod{m} \) takes \( O(\log(n) \log^2(m)) \).

**Extracting square roots mod \( p \):** to solve \( x^2 \equiv a \pmod{p} \) takes \( O(\log^4(p)) \) (using (RH)).

**Computations in finite fields:** Addition, multiplication and inverses/division in \( \mathbb{F}_q \) take \( O(\log^2(q)) \).

**Calculating** \( \phi(m) = |(\mathbb{Z}/m\mathbb{Z})^\times| \): (Sub)exponential; polynomial if the factorization of \( m \) is known.

**Calculating** \( \text{ord}(a \pmod{m}) \): Exponential; polynomial if the factorization of \( \phi(m) \) is known.

**Finding generators of** \( \mathbb{F}_p^\times \): Exponential; the random method works well if \( p-1 \) has few prime factors.