Arithmetik

- Berechnung von Potenzen
- euklidischer Algorithmus
- modulare Arithmetik
- RSA Verschlüsselung

Berechnung von Potenzen

Eingabe: $x \in \mathbb{C}$, $m \in \mathbb{N}$

Ausgabe: x^m

REPEATEDSQUARING(x, m)

if $m = 1$ **then return** x;
$y := x$; $z := 1$;
while $m > 1$ **do**
    if odd(m) **then** $z := z \cdot y$;
    $y := y^2$;
    $m := \lfloor \frac{m}{2} \rfloor$;
end
return z

Der euklidische Algorithmus

Eingabe: $a, b \in \mathbb{N}_0$, $a > 0$

Ausgabe: Tripel (ggT(a, b), α, β) mit $\alpha a + \beta b =$ ggT(a, b)

EUKLIDEXT(a, b)

$r_0 := a$; $r_1 := b$; $i := 1$;
$s_0 := 1$; $s_1 := 0$;
$t_0 := 0$; $t_1 := 1$;
while $r_i \neq 0$ **do**
    (* $q_i := r_{i-1}$ div r_i *)
    $q_i := \lfloor \frac{r_{i-1}}{r_i} \rfloor$;
    (* $r_{i+1} := r_{i-1}$ mod r_i *)
    $r_{i+1} := r_{i-1} - q_i r_i$;
    $s_{i+1} := s_{i-1} - q_i s_i$;
    $t_{i+1} := t_{i-1} - q_i t_i$;
    $i + +$;
end
return $(r_{i-1}, s_{i-1}, t_{i-1})$