Euklidischer Algorithmus
Proseminar Informatik "Algorithms Unplugged"
am Institut für Theoretische Informatik
Leibniz Universität Hannover

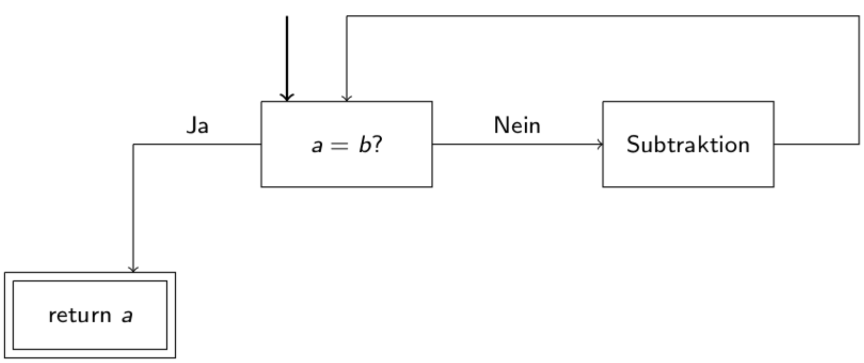
Bharat Ahuja
Wintersemester 2014/15

- Der euklidische Algorithmus ist ein Algorithmus, mit dem sich der größte gemeinsame Teiler (*g.g.T.*) zweier natürlicher Zahlen berechnen lässt.
- Diesen Algorithmus hat Euklid ca. 300 *v.Chr.* in *Buch VII – Die Elemente* (Proposition 1 und 2) als einen geometrischen Algorithmus vorgestellt.
- Berechnung von *g.g.T.* relevant bei Gruppierung von Objekten, Prüfen von Teilerfremdheit.
- Die Teilerfremdheit zweier Zahlen kann man alternativ durch das Vergleichen der Primfaktoren überprüfen.

  $$ggT(a,b) = \prod_{i \in \mathbb{N}} p_i^{min(a_i,b_i)}$$

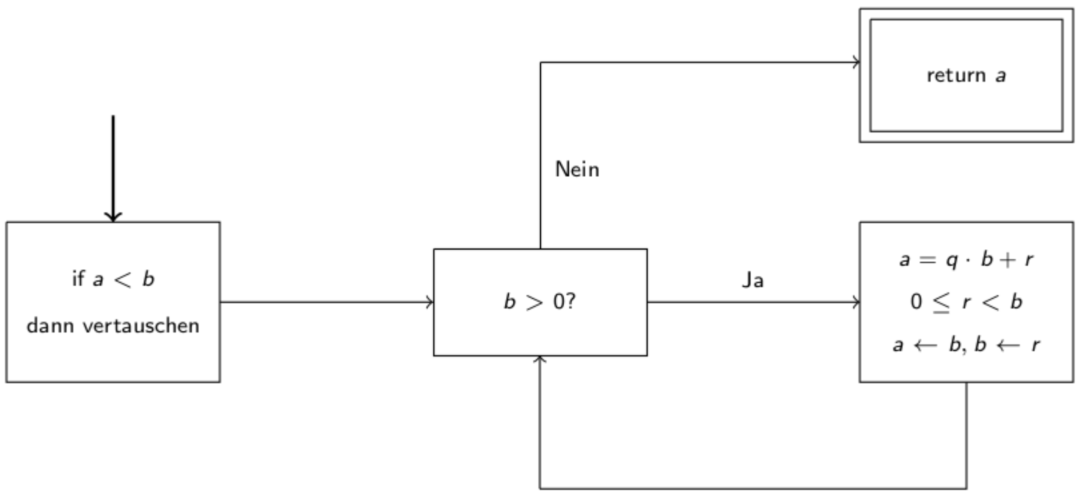
  wobei $p_i \in \mathbb{P}$.
- Die Bestimmung der Primfaktorzerlegung einer Zahl liegt aber in NP.
- LangsamEuklid

while $a \neq b$ **do**
    Falls a größer ist als $b, a \leftarrow a - b$
    Falls b größer ist als $a, b \leftarrow b - a$
end while
Gib den gemeinsamen Wert der Zahlen aus



- EUKLID

```
if a < b: vertausche a und b.
while b > 0 do
    berechne q, r mit a = q · b + r, wobei 0 ≤ r < b
    a ← b, b ← r
end while
return a
```



- EUKLID effizienter als LANGSAMEUKLID wegen Zusammenfassung mehrfacher Subtraktionen.

- Je schneller die Reste fallen, desto früher terminiert der Algorithmus. Die Fibonacci-Zahlen bilden den ungünstigsten Fall für diesen Algorithmus, weil alle Quotienten den kleinstmöglichen Wert annehmen, und dadurch die Reste sehr langsam fallen.

- Nach der ersten Iteration ist immer nur der Divisionsrest relevant.

$$\therefore T_b = \frac{1}{b} \sum_{0<k\leq b} T(k, b)$$

- Wenn man eine zufällig gewählte Zahl auf Teilerfremdheit mit 5 prüfen möchte, braucht man im Schnitt $T_5 = 2{,}6$ Iterationen. Es gilt,

$$T_n \approx \ln n + O(1)$$