

LU01.L02 - Imperativer ggT Algorithmus

Der Algorithmus zur Berechnung des größten gemeinsamen Teilers (GGT) von zwei Zahlen basiert auf dem euklidischen Algorithmus. Er ist eine effiziente Methode, den größten gemeinsamen Teiler zweier Zahlen zu finden. Hier ist die imperativ implementierte Musterlösung für den euklidischen Algorithmus in Python:

```
def ggt(a, b):  
    while b != 0:  
        cache = a  
        a = b  
        b = cache % b  
        # a, b = b, a % b Das ist der "pydantic" Ansatz um Variablen zu  
        vertauschen.  
    return a  
  
if __name__ == '__main__':  
    zahl1 = 56  
    zahl2 = 48  
    resultat = ggt(zahl1, zahl2)  
    print(f'Der GGT von {zahl1} und {zahl2} ist {resultat}') # Ausgabe: Der  
    GGT von 56 und 48 ist 8
```

Erklärung

1. **Schleife:** Der Algorithmus verwendet eine Schleife, die so lange läuft, wie b nicht Null ist. In jedem Schritt der Schleife wird der Wert von b mit dem Rest der Division von a durch b ersetzt, und der Wert von a wird mit dem ursprünglichen Wert von b ersetzt.
2. **Tausch:** In jedem Durchlauf der Schleife werden die Werte von a und b aktualisiert. Der Wert von a wird auf den Wert von b gesetzt, und der Wert von b wird auf den Rest der Division von a durch b gesetzt. Dieser Tausch wird in einer einzigen Zeile in Python durchgeführt.
3. **Ergebnis:** Wenn b Null wird, ist der aktuelle Wert von a der größte gemeinsame Teiler der beiden ursprünglichen Zahlen, und die Schleife wird beendet. Dieser Wert wird dann zurückgegeben.

Fazit

Der euklidische Algorithmus ist eine alte und effiziente Methode zur Berechnung des größten gemeinsamen Teilers. Er hat eine Laufzeitkomplexität von $O(\log(\min(a, b)))$, was ihn für die Anwendung auf sehr große Zahlen geeignet macht.

From:
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:
<https://wiki.bzz.ch/modul/m323/learningunits/lu01/loesungen/imperativereuklid>

Last update: **2024/03/28 14:07**

