

35. Wechselgeld, 10 Punkte

Pseudocode: Naive Idee, einfach formuliert: Eingabe: Betrag, Geldwerte()

1. sortiere Geldwerte absteigend, mach aus Geldwerte gleichzeitig Tupel (Geldwert, Anzahl)
2. Restbetrag = Betrag
3. für jeden Geldwert
 - (a) wenn $\text{Betrag} - \text{Geldwert} < 0$, dann nehme nächsten Geldwert
 - (b) ziehe solange Geldwert vom Restbetrag ab bis $\text{Restbetrag} < \text{Geldwert}$
4. ist der Restbetrag == 0, dann gebe verwendete Geldwerte zurück

Beste Lösung: nutze die mathematische Operation $\text{divmod}(x,y)$. **Code**

```
1 def money(b, values):
2     r = b
3     l = []
4     for k in sorted(values, reverse=True):
5         c, r = divmod(r, k)
6         l.append((k, c))
7     if r == 0:
8         a = str(b)+'='
9         for b in l:
10             for i in range(0, b[1]):
11                 a += str(b[0]) + '+'
12         print(a[0:len(a)-1])
13         return b, l
14     print('keine Lösung')
15     return b, []
```

Laufzeit: $n = \text{Geldwerte}$ (Länge der Liste)

- sortieren: $\mathcal{O}(n * \log n)$
- Werte berechnen: $\mathcal{O}(n)$
- Ausgabe: $\mathcal{O}(n * \text{Anzahl berechneter Geldwerte})$

Ohne die Ausgabe also $\mathcal{O}(n * \log n)$. Die Ausgabe, wie sie in der Aufgabenstellung vorhanden ist, kann die Laufzeit drastisch erhöhen.

38. Beste unabhängige Knotenmenge in Bäumen, 10 Punkte

Durchlaufe Baum ($G = (V,E)$) mit Tiefensuche und markiere (ggfs. teile gleich in getrennte Listen auf) die Knoten. Die folgende Eigenschaft: $\forall v, v' \in V'. v \neq v' \Rightarrow \{v, v'\} \notin E$ erfüllen.