

Spezialfall davon:

PROBLEM 1.9 ("PARTITION")

VL 3

24.04.2018

11

Gegeben:  $n$  Objekte  $1, \dots, n$  mit jeweils Größe  $z_i$

Gesucht:  $S \subseteq \{1, \dots, n\}$  mit

$$\sum_{i \in S} z_i = \sum_{i \notin S} z_i$$

1.2 Ein Beispiel

Beispiel 1.10

7, 13, 17, 20, 29, 31, 31, 35, 57

(Man erinnere sich an die allererste AuDI-Vorlesung!)

Gesamtsumme: 240

Also: Gibt es  $S \subseteq \{1, \dots, 9\}$  mit  $\sum_{i \in S} z_i = 120$ ?

Antwort: NEIN!

(Danks:

(7), 13, 17, 24, (29), (31), 31, 35, (53) )

Wie beweist man die Nichtexistenz?

(12)

Ziel: 120

Betrachte: 57 ✓ (o. B.d.A.)

(57)

Ziel: 63

Betrachte: 35 ?

Ziel: 28

Betrachte:

- 31 ×
- 31 ×
- 29 ×
- 25 ? → 8 ×
- 17 ? → 11 ×
- 13 ? → 15 ×
- 7 ? → 21 ×

Geht nicht!

(57)

(35)

Betrachte: 31 ?

Ziel: 32

Betrachte:

- 31 ×
- 29 ×
- 20 → 12 ×
- 17 → 15 ×
- 13 ×
- 7 ×

Geht nicht!

(57)

(35) (31) (31)

Betrachte: 29 ?

Ziel: 34

Betrachte:

- 20 → 14 ×
- 17 → 17 ×
- 13 ×
- 7 ×

Geht nicht!

(57)

(35) (31) (31) (29)  
→ 120

( Alternative:  $(35)(31) + (31) = 97$ , Ziel 23 )

13

geht nicht ... )

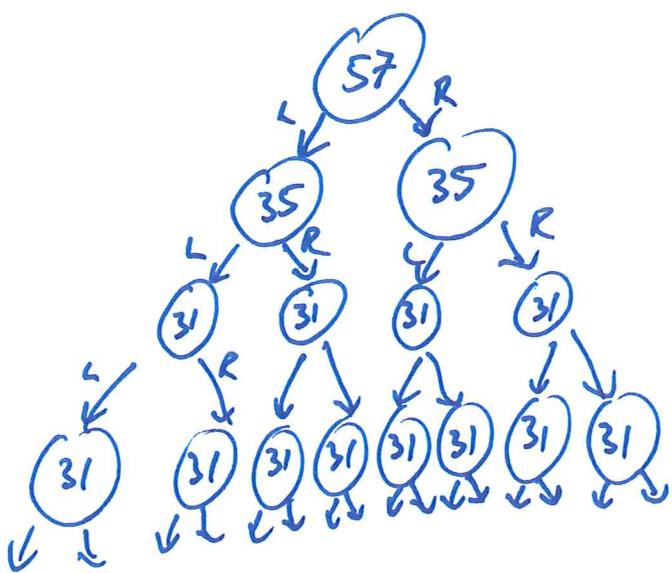
Also Idee:

ausprobieren, systematisch, von den größeren zu kleineren Zahlen  
 ↳ Rückwärtsrekursion über verwendete Zahlen und  
 erreichbare Werte!

Vorteil: Funktioniert

Nachteil: Probiert exponentiell viele Teilmengen durch

( Baumdarstellung:



$\downarrow$   
 $2^n$  Blätter!)

## Verbesserungen:

- Schneide Teilbäume ab wenn möglich  
 $\hookrightarrow$  z.T. Einsparungen, bleibt aber exponentiell
- Später: "Branch and Bound"

- Betrachte nicht Rückwärtsrekursion  
 (d.h. Probiere stufenweise aus, reduziere dabei größere Teilinstanzen auf kleinere Teilinstanzen)
- Sondern Vorwärtsrekursion:  
 Baue größere Lösungen aus kleineren Lösungen auf!

## BEISPIEL 1.10A

(xkcd 287)

(Herr)

### - Appetizers -

Mixed Fruit	2.15
French Fries	2.75
Side Salad	3.35
Hot Wings	3.55
Mozzarella Sticks	4.20
Sampler Plate	5.80

"We'd like exactly \$15.05 of appetizers, please."

"... exactly? Uh..."

"Here, these papers on the Knapsack problem might help you out."

"Listen, I have six other tables' to get to -"

"- as fast as possible, of course.  
Want something on Traveling Salesman?"

Fragen: - Wie lautet eine mögliche Kombination?

- Wie gut oder schlecht ist der Greedy-Algorithmus 1.4 für 0-1-Knapsack?
- Wie kann man den Algorithmus erweitern / modifizieren, dass er eine bessere Leistung generiert?

Große Fragen: - Wie bekommt man diese Probleme besser in den Griff?  
- Wie schwierig kann das werden - bzw. wie lange benötigt das Lösen?