

Aufgabe 5: Marktwaage

Team-ID: 60086

Team: JAB

Bearbeiter/-innen dieser Aufgabe:
Joshua Benning

7. September 2020

Inhaltsverzeichnis

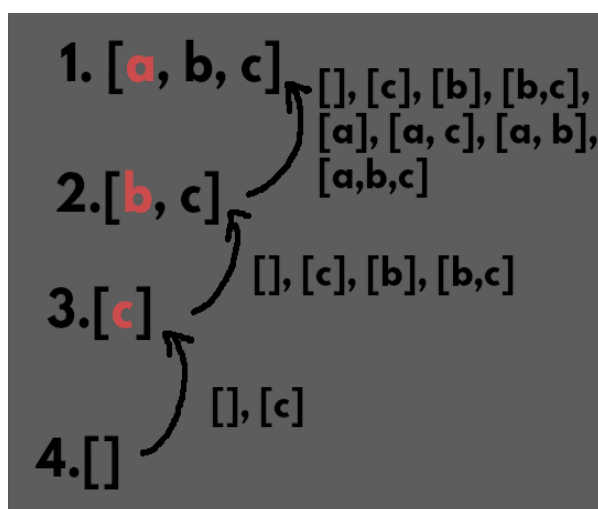
Lösungsidee.....	1
Umsetzung.....	1
Berechnung der Submengen:.....	1
Bestimmung der darstellbaren Gewichte:.....	2
Problematik des Lösungsansatzes.....	2
Beispiele.....	2
Beispiel 0.....	4
Beispiel 1.....	6
Beispiel 2.....	9
Beispiel 3.....	12
Beispiel 4.....	14
Beispiel 5.....	17
Quellcode.....	20

Lösungsidee

Ich berechne alle möglichen Submengen der Gewichte, also alle möglichen Kombinationen von Gewichten und versuche dann damit die gesuchten Gewichte darzustellen. Sollte eine Lösung nicht darstellbar sein, gebe ich die nächstmögliche Darstellung aus.

Umsetzung

Die Implementation erfolgt in Java. Das Programm besteht aus 2 Komponenten; der Berechnung der Submengen und der Ermittlung der darstellbaren Gewichte.



Berechnung der Submengen:

Zur Berechnung der Submengen wird ein rekursiver Ansatz gewählt um die sogenannte Potenzmenge zu bilden. Der Menge wird dabei immer das erste Element, der Kopf, entfernt und die Methode ruft sich selbst mit der verkürzten Liste auf. Ist die Menge

leer, gibt sie eine leere Liste zurück und die Rekursion bricht ab. Die Rückgabe der einzelnen Wiederholungen setzt sich aus zwei Komponenten zusammen. Einerseits wird die Rückgabe der folgenden Wiederholung übernommen, andererseits wird jeder Menge der Lösung der Kopf hinzugefügt. Wird also bei Ausführung 4 die leere Liste zurückgegeben, gibt die Ausführung 3 einmal die Ausgabe der vierten (leere Liste) unbearbeitet und zusätzlich die Ausgabe der vierten mit dem Kopf zurück (leere Liste + c).

Bestimmung der darstellbaren Gewichte:

Zur Bestimmung der darstellbaren Gewichte iteriere ich über die darzustellenden Gewichte und überprüfe, ob dieser darstellbar ist. Anfangs generiere ich alle Kombinationen der Gewichtsstücke (Berechnung der Submengen:) und deren Summen. Diese werden in der HashMap subMengen zwischengespeichert mit der Summe der Mengen als Schlüssel. Da der Schlüssel einzigartig sein muss werden dadurch bereits Mengen mit derselben Summe ausgeschlossen.

Daraufhin wird über die Gewichte iteriert und zwei Fälle überprüft:
1. Ob sich die Zahl als einfache Menge darstellen lässt

Und falls nicht ob

2. Die Zahl sich als Kombination aus zwei Summen darstellen lässt; Dabei wird über alle Submengen iteriert und überprüft, ob ihre Summe, summiert mit dem darzustellenden Gewicht der Summe, der Summe einer anderen Submenge entspricht. Ist dies der Fall wird überprüft, ob die dabei verwendeten Gewichte überhaupt vorliegen.

Danach wird erneut über die Gewichte iteriert und entweder die Summe direkt oder ein Verweis auf die nächste darstellbare Summe ausgegeben.

Problematik des Lösungsansatzes

Durch die Verwendung einer HashMap mit der Summe als Schlüssel, werden, wie bereits oben erwähnt, Kombinationen gleicher Summe ignoriert. Dadurch werden auch Zahlen als nicht direkt darstellbar angegeben, obwohl dies eventuell möglich wäre, mit anderen Gewichten. Die Lösung ist daher nicht vollständig und wird vor allem bei dem letzten Beispiel aufgrund der hohen Zahl an ignorierten Kombinationen nicht alle Zahlen korrekt darstellen. Bei den BWInf-Beispielen ist daher immer zusätzlich die Anzahl an ignorierten Kombinationen angegeben.

Beispiele

Beispiel zur Berechnung aller möglichen Kombinationen

Da die Größe der Potenzmenge $2^n(A)$ entspricht, wird hier als Beispiel $A=\{1,2,3\}$ verwendet um die Übersicht zu erhalten.

A entspricht hierbei der der Methode übergebenen Menge

R der Rückgabe der Methode

Der Menge wird bei jeder Iteration das erste Element entfernt und die Methode ruft sich selbst mit der verkürzten Liste auf.

$A=\{1,2,3\}$

$A_1=\{2,3\}$

$A_2 = \{3\}$

$A_3 = \{\}$

Ist die Menge leer, gibt die Methode eine leere Liste zurück und die Rekursion bricht ab. Die Rückgabe der einzelnen Wiederholungen setzt sich aus zwei Komponenten zusammen. Einerseits wird die Rückgabe der folgenden Wiederholung übernommen, andererseits wird jeder Menge der Lösung der Kopf hinzugefügt.

$R_{A_3} = \{\}$

$R_{A_2} = \{\{\}, \{3\}\}$

$R_{A_1} = \{\{\}, \{3\}, \{2\}, \{2,3\}\}$

$R_A = \{\{\}, \{3\}, \{2\}, \{2,3\}, \{1\}, \{1,3\}, \{1,2\}, \{1,2,3\}\}$

Man erhält die Potenzmenge und somit alle möglichen Kombinationen der Gewichte.

Beispiel zur Untersuchung der darzustellenden Gewichte

Basieren tut diese Methode auf dem Berechnen der Potenzmenge. Um auch hier die Übersicht zu behalten, verwende ich $A = \{1,2,5\}$ als Gewichtsstücke und $G = \{1,2,3,4,5\}$ als darzustellende Gewichte.

Zu Beginn wird die Potenzmenge berechnet und in einer HashMap mit der Summe einer Untermenge als Schlüssel abgebildet.

Am Beispiel:

$P_A = \{\{\}, \{4\}, \{2\}, \{2,4\}, \{1\}, \{1,4\}, \{1,2\}, \{1,2,4\}\}$

subMengen = {

1 = {1},

2 = {2},

3 = {1, 2},

5 = {5},

6 = {1, 5},

7 = {2, 5},

8 = {1, 2, 5}}

Nun wird über die darzustellenden Zahlen iteriert.

Sollte die darzustellende Zahl als Schlüssel in der HashMap stehen, hat man damit eine Lösung gefunden.

Am Beispiel: 1,2,3,5 lassen sich direkt darzustellen;

Lösung: 1 {} | {1}

Lösung: 2 {} | {2}

Lösung: 3 {} | {1, 2}

Lösung: 5 {} | {5}

Sollte sich eine Zahl allerdings nicht als Schlüssel in der HashMap befinden, wird über alle Einträge der HashMap iteriert und versucht, ob die Summe der Kombination plus die darzustellende Zahl der Summe einer anderen Kombination entspricht.

Am Beispiel: 4 ist kein Schlüssel in der HashMap;

Die Summe aus {1} und der darzustellenden Zahl 4 entspricht der Summe der Kombination {5}

Danach wird überprüft ob diese Lösung auch nicht die gegebenen Gewichte überschreitet. Dafür wird A kopiert. Aus dieser Kopie A_K werden dann alle links verwendeten Gewichte entfernt. Wenn sich dann alle rechts verwendeten Gewichte noch in A_K befinden, ist die Lösung valide.

BWInf-Beispiele

Beispiel 0

Summen gekürzt: 719

Summen ungekürzt: 32768

Links: 10 + [] | Rechts: [10]

Links: 20 + [] | Rechts: [10, 10]

Links: 30 + [] | Rechts: [10, 10, 10]

Links: 40 + [10] | Rechts: [50]

Links: 50 + [] | Rechts: [50]

Links: 60 + [] | Rechts: [10, 50]

Links: 70 + [] | Rechts: [10, 10, 50]

Links: 80 + [] | Rechts: [10, 10, 10, 50]

Links: 90 + [10] | Rechts: [50, 50]

Links: 100 + [] | Rechts: [50, 50]

Links: 110 + [] | Rechts: [10, 50, 50]

Links: 120 + [] | Rechts: [10, 10, 50, 50]

Links: 130 + [] | Rechts: [10, 10, 10, 50, 50]

Links: 140 + [10] | Rechts: [50, 100]

Links: 150 + [] | Rechts: [50, 100]

Links: 160 + [] | Rechts: [10, 50, 100]

Links: 170 + [] | Rechts: [10, 10, 50, 100]

Links: 180 + [] | Rechts: [10, 10, 10, 50, 100]

Links: 190 + [10] | Rechts: [50, 50, 100]

Links: 200 + [] | Rechts: [50, 50, 100]

Links: 210 + [] | Rechts: [10, 50, 50, 100]

Links: 220 + [] | Rechts: [10, 10, 50, 50, 100]

Links: 230 + [] | Rechts: [10, 10, 10, 50, 50, 100]

Links: 240 + [10] | Rechts: [50, 100, 100]

Links: 250 + [] | Rechts: [50, 100, 100]

Links: 260 + [] | Rechts: [10, 50, 100, 100]

Links: 270 + [] | Rechts: [10, 10, 50, 100, 100]

Links: 280 + [] | Rechts: [10, 10, 10, 50, 100, 100]

Links: 290 + [10] | Rechts: [50, 50, 100, 100]

Links: 300 + [] | Rechts: [50, 50, 100, 100]

Links: 310 + [] | Rechts: [10, 50, 50, 100, 100]

Links: 320 + [] | Rechts: [10, 10, 50, 50, 100, 100]
Links: 330 + [] | Rechts: [10, 10, 10, 50, 50, 100, 100]
Links: 340 + [10] | Rechts: [50, 100, 100, 100]
Links: 350 + [] | Rechts: [50, 100, 100, 100]
Links: 360 + [] | Rechts: [10, 50, 100, 100, 100]
Links: 370 + [] | Rechts: [10, 10, 50, 100, 100, 100]
Links: 380 + [] | Rechts: [10, 10, 10, 50, 100, 100, 100]
Links: 390 + [10] | Rechts: [50, 50, 100, 100, 100]
Links: 400 + [] | Rechts: [50, 50, 100, 100, 100]
Links: 410 + [] | Rechts: [10, 50, 50, 100, 100, 100]
Links: 420 + [] | Rechts: [10, 10, 50, 50, 100, 100, 100]
Links: 430 + [] | Rechts: [10, 10, 10, 50, 50, 100, 100, 100]
Links: 440 + [10, 50] | Rechts: [500]
Links: 450 + [50] | Rechts: [500]
Links: 460 + [50] | Rechts: [10, 500]
Links: 470 + [10, 10, 10] | Rechts: [500]
Links: 480 + [10, 10] | Rechts: [500]
Links: 490 + [10] | Rechts: [500]
Links: 500 + [] | Rechts: [500]
Links: 9500 + [] | Rechts: [500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9510 + [] | Rechts: [10, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9520 + [] | Rechts: [10, 10, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9530 + [] | Rechts: [10, 10, 10, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9540 + [10] | Rechts: [50, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9550 + [] | Rechts: [50, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9560 + [] | Rechts: [10, 50, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9570 + [] | Rechts: [10, 10, 50, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9580 + [] | Rechts: [10, 10, 10, 50, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9590 + [10] | Rechts: [50, 50, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9600 + [] | Rechts: [50, 50, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9610 + [] | Rechts: [10, 50, 50, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9620 + [] | Rechts: [10, 10, 50, 50, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9630 + [] | Rechts: [10, 10, 10, 50, 50, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9640 + [10] | Rechts: [50, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9650 + [] | Rechts: [50, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9660 + [] | Rechts: [10, 50, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9670 + [] | Rechts: [10, 10, 50, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9680 + [] | Rechts: [10, 10, 10, 50, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9690 + [10] | Rechts: [50, 50, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9700 + [] | Rechts: [50, 50, 100, 500, 500, 500, 1000, 1000, 1000, 5000]

Links: 9710 + [] | Rechts: [10, 50, 50, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9720 + [] | Rechts: [10, 10, 50, 50, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9730 + [] | Rechts: [10, 10, 10, 50, 50, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9740 + [10] | Rechts: [50, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9750 + [] | Rechts: [50, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9760 + [] | Rechts: [10, 50, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9770 + [] | Rechts: [10, 10, 50, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9780 + [] | Rechts: [10, 10, 10, 50, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9790 + [10] | Rechts: [50, 50, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9800 + [] | Rechts: [50, 50, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9810 + [] | Rechts: [10, 50, 50, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9820 + [] | Rechts: [10, 10, 50, 50, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9830 + [] | Rechts: [10, 10, 10, 50, 50, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9840 + [10] | Rechts: [50, 100, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9850 + [] | Rechts: [50, 100, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9860 + [] | Rechts: [10, 50, 100, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9870 + [] | Rechts: [10, 10, 50, 100, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9880 + [] | Rechts: [10, 10, 10, 50, 100, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9890 + [10] | Rechts: [50, 50, 100, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9900 + [] | Rechts: [50, 50, 100, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9910 + [] | Rechts: [10, 50, 50, 100, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9920 + [] | Rechts: [10, 10, 50, 50, 100, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Links: 9930 + [] | Rechts: [10, 10, 10, 50, 50, 100, 100, 100, 500, 500, 500, 1000, 1000, 1000, 5000]
Keine exakte Lösung zu 9940 | Nächste Lösung siehe 9930
Keine exakte Lösung zu 9950 | Nächste Lösung siehe 9930
Keine exakte Lösung zu 9960 | Nächste Lösung siehe 9930
Keine exakte Lösung zu 9970 | Nächste Lösung siehe 9930
Keine exakte Lösung zu 9980 | Nächste Lösung siehe 9930
Keine exakte Lösung zu 9990 | Nächste Lösung siehe 9930
Keine exakte Lösung zu 10000 | Nächste Lösung siehe 9930
Abdeckung: 99.3% - 993 Kombinationen
Benötigte Zeit: 0.11s

Beispiel 1

Summen gekürzt: 671

Summen ungekürzt: 131072

Links: 10 + [371] | Rechts: [127, 127, 127]

Keine exakte Lösung zu 20 | Nächste Lösung siehe 10

Keine exakte Lösung zu 30 | Nächste Lösung siehe 10

Keine exakte Lösung zu 40 | Nächste Lösung siehe 60

Keine exakte Lösung zu 50 | Nächste Lösung siehe 60
Links: 60 + [127, 371, 371, 371, 371, 371] | Rechts: [42, 2000]
Links: 70 + [371, 371, 371, 371, 371, 371] | Rechts: [42, 127, 127, 2000]
Keine exakte Lösung zu 80 | Nächste Lösung siehe 70
Keine exakte Lösung zu 90 | Nächste Lösung siehe 100
Links: 100 + [42, 42, 42, 2000] | Rechts: [371, 371, 371, 371, 371, 371]
Links: 110 + [42, 42, 42, 2000] | Rechts: [127, 127, 127, 371, 371, 371, 371, 371]
Keine exakte Lösung zu 120 | Nächste Lösung siehe 110
Keine exakte Lösung zu 130 | Nächste Lösung siehe 110
Keine exakte Lösung zu 140 | Nächste Lösung siehe 160
Keine exakte Lösung zu 150 | Nächste Lösung siehe 160
Links: 160 + [42, 42, 127] | Rechts: [371]
Links: 170 + [42, 42] | Rechts: [127, 127]
Keine exakte Lösung zu 180 | Nächste Lösung siehe 170
Keine exakte Lösung zu 190 | Nächste Lösung siehe 170
Keine exakte Lösung zu 200 | Nächste Lösung siehe 220
Keine exakte Lösung zu 210 | Nächste Lösung siehe 220
Links: 220 + [42, 127, 127, 371, 371, 371, 371] | Rechts: [2000]
Links: 230 + [42, 371, 371, 371, 371, 371] | Rechts: [127, 2000]
Keine exakte Lösung zu 240 | Nächste Lösung siehe 230
Keine exakte Lösung zu 250 | Nächste Lösung siehe 230
Keine exakte Lösung zu 260 | Nächste Lösung siehe 230
Keine exakte Lösung zu 270 | Nächste Lösung siehe 230
Keine exakte Lösung zu 280 | Nächste Lösung siehe 310
Keine exakte Lösung zu 290 | Nächste Lösung siehe 310
Keine exakte Lösung zu 300 | Nächste Lösung siehe 310
Links: 310 + [2000] | Rechts: [42, 42, 371, 371, 371, 371, 371, 371]
Links: 320 + [2000] | Rechts: [42, 42, 127, 127, 127, 371, 371, 371, 371, 371]
Keine exakte Lösung zu 330 | Nächste Lösung siehe 320
Keine exakte Lösung zu 340 | Nächste Lösung siehe 320
Keine exakte Lösung zu 350 | Nächste Lösung siehe 370
Keine exakte Lösung zu 360 | Nächste Lösung siehe 370
Links: 370 + [127] | Rechts: [42, 42, 42, 371]
Links: 380 + [] | Rechts: [42, 42, 42, 127, 127]
Links: 390 + [42, 42, 42, 371, 371, 371, 371] | Rechts: [2000]
Links: 400 + [42, 42, 42, 371, 371, 371, 371, 371] | Rechts: [127, 127, 127, 2000]
Keine exakte Lösung zu 410 | Nächste Lösung siehe 400
Keine exakte Lösung zu 420 | Nächste Lösung siehe 400
Keine exakte Lösung zu 430 | Nächste Lösung siehe 400

Keine exakte Lösung zu 440 | Nächste Lösung siehe 400

Keine exakte Lösung zu 450 | Nächste Lösung siehe 480

Keine exakte Lösung zu 460 | Nächste Lösung siehe 480

Keine exakte Lösung zu 470 | Nächste Lösung siehe 480

Links: 480 + [2000] | Rechts: [127, 127, 371, 371, 371, 371, 371, 371]

Keine exakte Lösung zu 490 | Nächste Lösung siehe 480

Keine exakte Lösung zu 500 | Nächste Lösung siehe 480

Keine exakte Lösung zu 9500 | Nächste Lösung siehe 9470

Keine exakte Lösung zu 9510 | Nächste Lösung siehe 9470

Keine exakte Lösung zu 9520 | Nächste Lösung siehe 9470

Keine exakte Lösung zu 9530 | Nächste Lösung siehe 9470

Keine exakte Lösung zu 9540 | Nächste Lösung siehe 9600

Keine exakte Lösung zu 9550 | Nächste Lösung siehe 9600

Keine exakte Lösung zu 9560 | Nächste Lösung siehe 9600

Keine exakte Lösung zu 9570 | Nächste Lösung siehe 9600

Keine exakte Lösung zu 9580 | Nächste Lösung siehe 9600

Keine exakte Lösung zu 9590 | Nächste Lösung siehe 9600

Links: 9600 + [127, 127, 127] | Rechts: [42, 42, 42, 371, 371, 371, 371, 371, 2000, 2000, 2000, 2000]

Links: 9610 + [] | Rechts: [42, 42, 42, 371, 371, 371, 371, 2000, 2000, 2000, 2000]

Links: 9620 + [] | Rechts: [42, 42, 42, 127, 127, 127, 371, 371, 371, 2000, 2000, 2000, 2000]

Links: 9630 + [42, 42, 42, 371] | Rechts: [127, 2000, 2000, 2000, 2000, 2000]

Keine exakte Lösung zu 9640 | Nächste Lösung siehe 9630

Keine exakte Lösung zu 9650 | Nächste Lösung siehe 9630

Keine exakte Lösung zu 9660 | Nächste Lösung siehe 9630

Keine exakte Lösung zu 9670 | Nächste Lösung siehe 9630

Keine exakte Lösung zu 9680 | Nächste Lösung siehe 9630

Keine exakte Lösung zu 9690 | Nächste Lösung siehe 9630

Keine exakte Lösung zu 9700 | Nächste Lösung siehe 9630

Keine exakte Lösung zu 9710 | Nächste Lösung siehe 9770

Keine exakte Lösung zu 9720 | Nächste Lösung siehe 9770

Keine exakte Lösung zu 9730 | Nächste Lösung siehe 9770

Keine exakte Lösung zu 9740 | Nächste Lösung siehe 9770

Keine exakte Lösung zu 9750 | Nächste Lösung siehe 9770

Keine exakte Lösung zu 9760 | Nächste Lösung siehe 9770

Links: 9770 + [127] | Rechts: [42, 371, 371, 371, 371, 371, 2000, 2000, 2000, 2000]

Links: 9780 + [] | Rechts: [42, 127, 127, 371, 371, 371, 371, 2000, 2000, 2000, 2000]

Keine exakte Lösung zu 9790 | Nächste Lösung siehe 9780

Keine exakte Lösung zu 9800 | Nächste Lösung siehe 9780

Keine exakte Lösung zu 9810 | Nächste Lösung siehe 9830

Keine exakte Lösung zu 9820 | Nächste Lösung siehe 9830

Links: 9830 + [127, 127] | Rechts: [42, 42, 2000, 2000, 2000, 2000, 2000]

Links: 9840 + [371] | Rechts: [42, 42, 127, 2000, 2000, 2000, 2000, 2000]

Keine exakte Lösung zu 9850 | Nächste Lösung siehe 9840

Keine exakte Lösung zu 9860 | Nächste Lösung siehe 9840

Keine exakte Lösung zu 9870 | Nächste Lösung siehe 9840

Keine exakte Lösung zu 9880 | Nächste Lösung siehe 9840

Keine exakte Lösung zu 9890 | Nächste Lösung siehe 9930

Keine exakte Lösung zu 9900 | Nächste Lösung siehe 9930

Keine exakte Lösung zu 9910 | Nächste Lösung siehe 9930

Keine exakte Lösung zu 9920 | Nächste Lösung siehe 9930

Links: 9930 + [42, 127, 127] | Rechts: [371, 371, 371, 371, 371, 371, 2000, 2000, 2000, 2000]

Links: 9940 + [42] | Rechts: [127, 371, 371, 371, 371, 371, 2000, 2000, 2000, 2000]

Keine exakte Lösung zu 9950 | Nächste Lösung siehe 9940

Keine exakte Lösung zu 9960 | Nächste Lösung siehe 9940

Keine exakte Lösung zu 9970 | Nächste Lösung siehe 9990

Keine exakte Lösung zu 9980 | Nächste Lösung siehe 9990

Links: 9990 + [127, 127, 127] | Rechts: [371, 2000, 2000, 2000, 2000, 2000]

Links: 10000 + [] | Rechts: [2000, 2000, 2000, 2000, 2000]

Abdeckung: 30.0% - 300 Kombinationen

Benötigte Zeit: 0.293s

Beispiel 2

Summen gekürzt: 1023

Summen ungekürzt: 1024

Links: 10 + [] | Rechts: [10]

Links: 20 + [] | Rechts: [20]

Links: 30 + [] | Rechts: [10, 20]

Links: 40 + [] | Rechts: [40]

Links: 50 + [] | Rechts: [10, 40]

Links: 60 + [] | Rechts: [20, 40]

Links: 70 + [] | Rechts: [10, 20, 40]

Links: 80 + [] | Rechts: [80]

Links: 90 + [] | Rechts: [10, 80]

Links: 100 + [] | Rechts: [20, 80]

Links: 110 + [] | Rechts: [10, 20, 80]

Links: 120 + [] | Rechts: [40, 80]

Links: 130 + [] | Rechts: [10, 40, 80]

Links: 140 + [] | Rechts: [20, 40, 80]

Links: 150 + [] | Rechts: [10, 20, 40, 80]

Links: 160 + [] | Rechts: [160]

Links: 170 + [] | Rechts: [10, 160]
Links: 180 + [] | Rechts: [20, 160]
Links: 190 + [] | Rechts: [10, 20, 160]
Links: 200 + [] | Rechts: [40, 160]
Links: 210 + [] | Rechts: [10, 40, 160]
Links: 220 + [] | Rechts: [20, 40, 160]
Links: 230 + [] | Rechts: [10, 20, 40, 160]
Links: 240 + [] | Rechts: [80, 160]
Links: 250 + [] | Rechts: [10, 80, 160]
Links: 260 + [] | Rechts: [20, 80, 160]
Links: 270 + [] | Rechts: [10, 20, 80, 160]
Links: 280 + [] | Rechts: [40, 80, 160]
Links: 290 + [] | Rechts: [10, 40, 80, 160]
Links: 300 + [] | Rechts: [20, 40, 80, 160]
Links: 310 + [] | Rechts: [10, 20, 40, 80, 160]
Links: 320 + [] | Rechts: [320]
Links: 330 + [] | Rechts: [10, 320]
Links: 340 + [] | Rechts: [20, 320]
Links: 350 + [] | Rechts: [10, 20, 320]
Links: 360 + [] | Rechts: [40, 320]
Links: 370 + [] | Rechts: [10, 40, 320]
Links: 380 + [] | Rechts: [20, 40, 320]
Links: 390 + [] | Rechts: [10, 20, 40, 320]
Links: 400 + [] | Rechts: [80, 320]
Links: 410 + [] | Rechts: [10, 80, 320]
Links: 420 + [] | Rechts: [20, 80, 320]
Links: 430 + [] | Rechts: [10, 20, 80, 320]
Links: 440 + [] | Rechts: [40, 80, 320]
Links: 450 + [] | Rechts: [10, 40, 80, 320]
Links: 460 + [] | Rechts: [20, 40, 80, 320]
Links: 470 + [] | Rechts: [10, 20, 40, 80, 320]
Links: 480 + [] | Rechts: [160, 320]
Links: 490 + [] | Rechts: [10, 160, 320]
Links: 500 + [] | Rechts: [20, 160, 320]
Links: 9500 + [] | Rechts: [20, 40, 160, 320, 1280, 2560, 5120]
Links: 9510 + [] | Rechts: [10, 20, 40, 160, 320, 1280, 2560, 5120]
Links: 9520 + [] | Rechts: [80, 160, 320, 1280, 2560, 5120]
Links: 9530 + [] | Rechts: [10, 80, 160, 320, 1280, 2560, 5120]
Links: 9540 + [] | Rechts: [20, 80, 160, 320, 1280, 2560, 5120]
Links: 9550 + [] | Rechts: [10, 20, 80, 160, 320, 1280, 2560, 5120]

Links: 9560 + [] | Rechts: [40, 80, 160, 320, 1280, 2560, 5120]
Links: 9570 + [] | Rechts: [10, 40, 80, 160, 320, 1280, 2560, 5120]
Links: 9580 + [] | Rechts: [20, 40, 80, 160, 320, 1280, 2560, 5120]
Links: 9590 + [] | Rechts: [10, 20, 40, 80, 160, 320, 1280, 2560, 5120]
Links: 9600 + [] | Rechts: [640, 1280, 2560, 5120]
Links: 9610 + [] | Rechts: [10, 640, 1280, 2560, 5120]
Links: 9620 + [] | Rechts: [20, 640, 1280, 2560, 5120]
Links: 9630 + [] | Rechts: [10, 20, 640, 1280, 2560, 5120]
Links: 9640 + [] | Rechts: [40, 640, 1280, 2560, 5120]
Links: 9650 + [] | Rechts: [10, 40, 640, 1280, 2560, 5120]
Links: 9660 + [] | Rechts: [20, 40, 640, 1280, 2560, 5120]
Links: 9670 + [] | Rechts: [10, 20, 40, 640, 1280, 2560, 5120]
Links: 9680 + [] | Rechts: [80, 640, 1280, 2560, 5120]
Links: 9690 + [] | Rechts: [10, 80, 640, 1280, 2560, 5120]
Links: 9700 + [] | Rechts: [20, 80, 640, 1280, 2560, 5120]
Links: 9710 + [] | Rechts: [10, 20, 80, 640, 1280, 2560, 5120]
Links: 9720 + [] | Rechts: [40, 80, 640, 1280, 2560, 5120]
Links: 9730 + [] | Rechts: [10, 40, 80, 640, 1280, 2560, 5120]
Links: 9740 + [] | Rechts: [20, 40, 80, 640, 1280, 2560, 5120]
Links: 9750 + [] | Rechts: [10, 20, 40, 80, 640, 1280, 2560, 5120]
Links: 9760 + [] | Rechts: [160, 640, 1280, 2560, 5120]
Links: 9770 + [] | Rechts: [10, 160, 640, 1280, 2560, 5120]
Links: 9780 + [] | Rechts: [20, 160, 640, 1280, 2560, 5120]
Links: 9790 + [] | Rechts: [10, 20, 160, 640, 1280, 2560, 5120]
Links: 9800 + [] | Rechts: [40, 160, 640, 1280, 2560, 5120]
Links: 9810 + [] | Rechts: [10, 40, 160, 640, 1280, 2560, 5120]
Links: 9820 + [] | Rechts: [20, 40, 160, 640, 1280, 2560, 5120]
Links: 9830 + [] | Rechts: [10, 20, 40, 160, 640, 1280, 2560, 5120]
Links: 9840 + [] | Rechts: [80, 160, 640, 1280, 2560, 5120]
Links: 9850 + [] | Rechts: [10, 80, 160, 640, 1280, 2560, 5120]
Links: 9860 + [] | Rechts: [20, 80, 160, 640, 1280, 2560, 5120]
Links: 9870 + [] | Rechts: [10, 20, 80, 160, 640, 1280, 2560, 5120]
Links: 9880 + [] | Rechts: [40, 80, 160, 640, 1280, 2560, 5120]
Links: 9890 + [] | Rechts: [10, 40, 80, 160, 640, 1280, 2560, 5120]
Links: 9900 + [] | Rechts: [20, 40, 80, 160, 640, 1280, 2560, 5120]
Links: 9910 + [] | Rechts: [10, 20, 40, 80, 160, 640, 1280, 2560, 5120]
Links: 9920 + [] | Rechts: [320, 640, 1280, 2560, 5120]
Links: 9930 + [] | Rechts: [10, 320, 640, 1280, 2560, 5120]
Links: 9940 + [] | Rechts: [20, 320, 640, 1280, 2560, 5120]
Links: 9950 + [] | Rechts: [10, 20, 320, 640, 1280, 2560, 5120]

Links: 9960 + [] | Rechts: [40, 320, 640, 1280, 2560, 5120]

Links: 9970 + [] | Rechts: [10, 40, 320, 640, 1280, 2560, 5120]

Links: 9980 + [] | Rechts: [20, 40, 320, 640, 1280, 2560, 5120]

Links: 9990 + [] | Rechts: [10, 20, 40, 320, 640, 1280, 2560, 5120]

Links: 10000 + [] | Rechts: [80, 320, 640, 1280, 2560, 5120]

Abdeckung: 100.0% - 1000 Kombinationen

Benötigte Zeit: 0.063s

Beispiel 3

Summen gekürzt: 127

Summen ungekürzt: 128

Lösung: Links: 10 + [] | Rechts: [10]

Lösung: Links: 20 + [10] | Rechts: [30]

Lösung: Links: 30 + [] | Rechts: [30]

Lösung: Links: 40 + [] | Rechts: [10, 30]

Lösung: Links: 50 + [10, 30] | Rechts: [90]

Lösung: Links: 60 + [30] | Rechts: [90]

Lösung: Links: 70 + [30] | Rechts: [10, 90]

Lösung: Links: 80 + [10] | Rechts: [90]

Lösung: Links: 90 + [] | Rechts: [90]

Lösung: Links: 100 + [] | Rechts: [10, 90]

Lösung: Links: 110 + [10] | Rechts: [30, 90]

Lösung: Links: 120 + [] | Rechts: [30, 90]

Lösung: Links: 130 + [] | Rechts: [10, 30, 90]

Lösung: Links: 140 + [10, 30, 90] | Rechts: [270]

Lösung: Links: 150 + [30, 90] | Rechts: [270]

Lösung: Links: 160 + [30, 90] | Rechts: [10, 270]

Lösung: Links: 170 + [10, 90] | Rechts: [270]

Lösung: Links: 180 + [90] | Rechts: [270]

Lösung: Links: 190 + [90] | Rechts: [10, 270]

Lösung: Links: 200 + [10, 90] | Rechts: [30, 270]

Lösung: Links: 210 + [90] | Rechts: [30, 270]

Lösung: Links: 220 + [90] | Rechts: [10, 30, 270]

Lösung: Links: 230 + [10, 30] | Rechts: [270]

Lösung: Links: 240 + [30] | Rechts: [270]

Lösung: Links: 250 + [30] | Rechts: [10, 270]

Lösung: Links: 260 + [10] | Rechts: [270]

Lösung: Links: 270 + [] | Rechts: [270]

Lösung: Links: 280 + [] | Rechts: [10, 270]

Lösung: Links: $290 + [10]$ | Rechts: $[30, 270]$

Lösung: Links: $300 + []$ | Rechts: $[30, 270]$

Lösung: Links: $310 + []$ | Rechts: $[10, 30, 270]$

Lösung: Links: $320 + [10, 30]$ | Rechts: $[90, 270]$

Lösung: Links: $330 + [30]$ | Rechts: $[90, 270]$

Lösung: Links: $340 + [30]$ | Rechts: $[10, 90, 270]$

Lösung: Links: $350 + [10]$ | Rechts: $[90, 270]$

Lösung: Links: $360 + []$ | Rechts: $[90, 270]$

Lösung: Links: $370 + []$ | Rechts: $[10, 90, 270]$

Lösung: Links: $380 + [10]$ | Rechts: $[30, 90, 270]$

Lösung: Links: $390 + []$ | Rechts: $[30, 90, 270]$

Lösung: Links: $400 + []$ | Rechts: $[10, 30, 90, 270]$

Lösung: Links: $410 + [10, 30, 90, 270]$ | Rechts: $[810]$

Lösung: Links: $420 + [30, 90, 270]$ | Rechts: $[810]$

Lösung: Links: $430 + [30, 90, 270]$ | Rechts: $[10, 810]$

Lösung: Links: $440 + [10, 90, 270]$ | Rechts: $[810]$

Lösung: Links: $450 + [90, 270]$ | Rechts: $[810]$

Lösung: Links: $460 + [90, 270]$ | Rechts: $[10, 810]$

Lösung: Links: $470 + [10, 90, 270]$ | Rechts: $[30, 810]$

Lösung: Links: $480 + [90, 270]$ | Rechts: $[30, 810]$

Lösung: Links: $490 + [90, 270]$ | Rechts: $[10, 30, 810]$

Lösung: Links: $500 + [10, 30, 270]$ | Rechts: $[810]$

Lösung: Links: $9500 + [10, 30, 270]$ | Rechts: $[90, 2430, 7290]$

Lösung: Links: $9510 + [30, 270]$ | Rechts: $[90, 2430, 7290]$

Lösung: Links: $9520 + [30, 270]$ | Rechts: $[10, 90, 2430, 7290]$

Lösung: Links: $9530 + [10, 270]$ | Rechts: $[90, 2430, 7290]$

Lösung: Links: $9540 + [270]$ | Rechts: $[90, 2430, 7290]$

Lösung: Links: $9550 + [270]$ | Rechts: $[10, 90, 2430, 7290]$

Lösung: Links: $9560 + [10, 270]$ | Rechts: $[30, 90, 2430, 7290]$

Lösung: Links: $9570 + [270]$ | Rechts: $[30, 90, 2430, 7290]$

Lösung: Links: $9580 + [270]$ | Rechts: $[10, 30, 90, 2430, 7290]$

Lösung: Links: $9590 + [10, 30, 90]$ | Rechts: $[2430, 7290]$

Lösung: Links: $9600 + [30, 90]$ | Rechts: $[2430, 7290]$

Lösung: Links: $9610 + [30, 90]$ | Rechts: $[10, 2430, 7290]$

Lösung: Links: $9620 + [10, 90]$ | Rechts: $[2430, 7290]$

Lösung: Links: $9630 + [90]$ | Rechts: $[2430, 7290]$

Lösung: Links: $9640 + [90]$ | Rechts: $[10, 2430, 7290]$

Lösung: Links: $9650 + [10, 90]$ | Rechts: $[30, 2430, 7290]$

Lösung: Links: $9660 + [90]$ | Rechts: $[30, 2430, 7290]$

Lösung: Links: 9670 + [90] | Rechts: [10, 30, 2430, 7290]
Lösung: Links: 9680 + [10, 30] | Rechts: [2430, 7290]
Lösung: Links: 9690 + [30] | Rechts: [2430, 7290]
Lösung: Links: 9700 + [30] | Rechts: [10, 2430, 7290]
Lösung: Links: 9710 + [10] | Rechts: [2430, 7290]
Lösung: Links: 9720 + [] | Rechts: [2430, 7290]
Lösung: Links: 9730 + [] | Rechts: [10, 2430, 7290]
Lösung: Links: 9740 + [10] | Rechts: [30, 2430, 7290]
Lösung: Links: 9750 + [] | Rechts: [30, 2430, 7290]
Lösung: Links: 9760 + [] | Rechts: [10, 30, 2430, 7290]
Lösung: Links: 9770 + [10, 30] | Rechts: [90, 2430, 7290]
Lösung: Links: 9780 + [30] | Rechts: [90, 2430, 7290]
Lösung: Links: 9790 + [30] | Rechts: [10, 90, 2430, 7290]
Lösung: Links: 9800 + [10] | Rechts: [90, 2430, 7290]
Lösung: Links: 9810 + [] | Rechts: [90, 2430, 7290]
Lösung: Links: 9820 + [] | Rechts: [10, 90, 2430, 7290]
Lösung: Links: 9830 + [10] | Rechts: [30, 90, 2430, 7290]
Lösung: Links: 9840 + [] | Rechts: [30, 90, 2430, 7290]
Lösung: Links: 9850 + [] | Rechts: [10, 30, 90, 2430, 7290]
Lösung: Links: 9860 + [10, 30, 90] | Rechts: [270, 2430, 7290]
Lösung: Links: 9870 + [30, 90] | Rechts: [270, 2430, 7290]
Lösung: Links: 9880 + [30, 90] | Rechts: [10, 270, 2430, 7290]
Lösung: Links: 9890 + [10, 90] | Rechts: [270, 2430, 7290]
Lösung: Links: 9900 + [90] | Rechts: [270, 2430, 7290]
Lösung: Links: 9910 + [90] | Rechts: [10, 270, 2430, 7290]
Lösung: Links: 9920 + [10, 90] | Rechts: [30, 270, 2430, 7290]
Lösung: Links: 9930 + [90] | Rechts: [30, 270, 2430, 7290]
Lösung: Links: 9940 + [90] | Rechts: [10, 30, 270, 2430, 7290]
Lösung: Links: 9950 + [10, 30] | Rechts: [270, 2430, 7290]
Lösung: Links: 9960 + [30] | Rechts: [270, 2430, 7290]
Lösung: Links: 9970 + [30] | Rechts: [10, 270, 2430, 7290]
Lösung: Links: 9980 + [10] | Rechts: [270, 2430, 7290]
Lösung: Links: 9990 + [] | Rechts: [270, 2430, 7290]
Lösung: Links: 10000 + [] | Rechts: [10, 270, 2430, 7290]
Abdeckung: 100.0% - 1000 Kombinationen
Benötigte Zeit: 0.083s

Beispiel 4

Summen gekürzt: 959

Summen ungekürzt: 8192

Links: 10 + [29, 29] | Rechts: [5, 21, 21, 21]
Links: 20 + [29, 259] | Rechts: [21, 287]
Links: 30 + [287] | Rechts: [29, 29, 259]
Links: 40 + [5, 21, 21] | Rechts: [29, 29, 29]
Links: 50 + [] | Rechts: [21, 29]
Links: 60 + [5, 21, 259] | Rechts: [29, 29, 287]
Links: 70 + [259] | Rechts: [21, 21, 287]
Links: 80 + [287] | Rechts: [21, 29, 29, 29, 259]
Links: 90 + [21, 29, 259] | Rechts: [399]
Links: 100 + [] | Rechts: [21, 21, 29, 29]
Links: 110 + [5, 259] | Rechts: [29, 29, 29, 287]
Links: 120 + [259] | Rechts: [21, 21, 21, 29, 287]
Links: 130 + [21, 21, 29, 29, 29] | Rechts: [259]
Links: 140 + [259] | Rechts: [399]
Links: 150 + [] | Rechts: [21, 21, 21, 29, 29, 29]
Links: 160 + [21, 399] | Rechts: [5, 29, 259, 287]
Links: 170 + [287] | Rechts: [29, 29, 399]
Links: 180 + [21, 29, 29] | Rechts: [259]
Links: 190 + [5, 21, 21, 21, 29] | Rechts: [287]
Links: 200 + [29, 29, 29] | Rechts: [287]
Links: 210 + [399] | Rechts: [21, 21, 21, 259, 287]
Links: 220 + [5, 21, 21, 21] | Rechts: [29, 259]
Links: 230 + [29] | Rechts: [259]
Links: 240 + [5, 21, 21] | Rechts: [287]
Links: 250 + [21, 21] | Rechts: [5, 287]
Links: 260 + [399] | Rechts: [5, 21, 29, 29, 29, 259, 287]
Links: 270 + [5, 21, 21] | Rechts: [29, 29, 259]
Links: 280 + [] | Rechts: [21, 259]
Links: 290 + [5, 21] | Rechts: [29, 287]
Links: 300 + [21] | Rechts: [5, 29, 287]
Links: 310 + [29, 29, 29, 287] | Rechts: [5, 21, 259, 399]
Links: 320 + [5, 21] | Rechts: [29, 29, 29, 259]
Links: 330 + [] | Rechts: [21, 21, 29, 259]
Links: 340 + [5] | Rechts: [29, 29, 287]
Links: 350 + [] | Rechts: [21, 21, 21, 287]
Links: 360 + [5, 21, 21, 21] | Rechts: [29, 399]
Links: 370 + [29] | Rechts: [399]
Links: 380 + [] | Rechts: [21, 21, 21, 29, 29, 259]

Links: 390 + [5] | Rechts: [21, 29, 29, 29, 287]
Links: 400 + [] | Rechts: [5, 21, 29, 29, 29, 287]
Links: 410 + [5, 21, 21] | Rechts: [29, 29, 399]
Links: 420 + [] | Rechts: [21, 399]
Links: 430 + [21, 21, 21, 29, 29] | Rechts: [5, 259, 287]
Links: 440 + [21, 259] | Rechts: [5, 29, 287, 399]
Links: 450 + [287] | Rechts: [21, 29, 29, 259, 399]
Links: 460 + [5, 21] | Rechts: [29, 29, 29, 399]
Links: 470 + [] | Rechts: [21, 21, 29, 399]
Links: 480 + [21, 21, 29] | Rechts: [5, 259, 287]
Links: 490 + [259] | Rechts: [21, 21, 21, 287, 399]
Links: 500 + [287] | Rechts: [21, 21, 29, 29, 29, 259, 399]
Links: 9500 + [287] | Rechts: [21, 21, 21, 29, 29, 29, 259, 399, 2993, 2993, 2993]
Links: 9510 + [5, 21, 21, 21, 29, 29, 29] | Rechts: [287, 399, 2993, 2993, 2993]
Links: 9520 + [5] | Rechts: [259, 287, 2993, 2993, 2993]
Links: 9530 + [] | Rechts: [5, 259, 287, 2993, 2993, 2993]
Links: 9540 + [5, 21, 21, 21, 29] | Rechts: [259, 399, 2993, 2993, 2993]
Links: 9550 + [29, 29, 29] | Rechts: [259, 399, 2993, 2993, 2993]
Links: 9560 + [5, 21, 21, 29, 29] | Rechts: [287, 399, 2993, 2993, 2993]
Links: 9570 + [5] | Rechts: [21, 29, 259, 287, 2993, 2993, 2993]
Links: 9580 + [] | Rechts: [5, 21, 29, 259, 287, 2993, 2993, 2993]
Links: 9590 + [5, 21, 21] | Rechts: [259, 399, 2993, 2993, 2993]
Links: 9600 + [21, 21] | Rechts: [5, 259, 399, 2993, 2993, 2993]
Links: 9610 + [5, 21, 29] | Rechts: [287, 399, 2993, 2993, 2993]
Links: 9620 + [5] | Rechts: [21, 21, 29, 29, 259, 287, 2993, 2993, 2993]
Links: 9630 + [] | Rechts: [5, 21, 21, 29, 29, 259, 287, 2993, 2993, 2993]
Links: 9640 + [5, 21] | Rechts: [29, 259, 399, 2993, 2993, 2993]
Links: 9650 + [21] | Rechts: [5, 29, 259, 399, 2993, 2993, 2993]
Links: 9660 + [5] | Rechts: [287, 399, 2993, 2993, 2993]
Links: 9670 + [] | Rechts: [5, 287, 399, 2993, 2993, 2993]
Links: 9680 + [] | Rechts: [5, 21, 21, 21, 29, 29, 29, 259, 287, 2993, 2993, 2993]
Links: 9690 + [5] | Rechts: [29, 29, 259, 399, 2993, 2993, 2993]
Links: 9700 + [] | Rechts: [21, 21, 21, 259, 399, 2993, 2993, 2993]
Links: 9710 + [5] | Rechts: [21, 29, 287, 399, 2993, 2993, 2993]
Links: 9720 + [] | Rechts: [5, 21, 29, 287, 399, 2993, 2993, 2993]
Keine exakte Lösung zu 9730 | Nächste Lösung siehe 9720
Links: 9740 + [5] | Rechts: [21, 29, 29, 29, 259, 399, 2993, 2993, 2993]

Links: 9750 + [] | Rechts: [5, 21, 29, 29, 29, 259, 399, 2993, 2993, 2993]

Links: 9760 + [5] | Rechts: [21, 21, 29, 29, 287, 399, 2993, 2993, 2993]

Links: 9770 + [] | Rechts: [5, 21, 21, 29, 29, 287, 399, 2993, 2993, 2993]

Keine exakte Lösung zu 9780 | Nächste Lösung siehe 9770

Links: 9790 + [5, 21, 21, 29, 29, 29] | Rechts: [259, 287, 399, 2993, 2993, 2993]

Links: 9800 + [21, 21, 29, 29, 29] | Rechts: [5, 259, 287, 399, 2993, 2993, 2993]

Links: 9810 + [5] | Rechts: [21, 21, 21, 29, 29, 29, 287, 399, 2993, 2993, 2993]

Links: 9820 + [] | Rechts: [5, 21, 21, 21, 29, 29, 29, 287, 399, 2993, 2993, 2993]

Keine exakte Lösung zu 9830 | Nächste Lösung siehe 9820

Links: 9840 + [5, 21, 29, 29] | Rechts: [259, 287, 399, 2993, 2993, 2993]

Links: 9850 + [21, 29, 29] | Rechts: [5, 259, 287, 399, 2993, 2993, 2993]

Keine exakte Lösung zu 9860 | Nächste Lösung siehe 9850

Keine exakte Lösung zu 9870 | Nächste Lösung siehe 9850

Keine exakte Lösung zu 9880 | Nächste Lösung siehe 9890

Links: 9890 + [5, 29] | Rechts: [259, 287, 399, 2993, 2993, 2993]

Links: 9900 + [29] | Rechts: [5, 259, 287, 399, 2993, 2993, 2993]

Keine exakte Lösung zu 9910 | Nächste Lösung siehe 9900

Keine exakte Lösung zu 9920 | Nächste Lösung siehe 9900

Keine exakte Lösung zu 9930 | Nächste Lösung siehe 9940

Links: 9940 + [5] | Rechts: [21, 259, 287, 399, 2993, 2993, 2993]

Links: 9950 + [] | Rechts: [5, 21, 259, 287, 399, 2993, 2993, 2993]

Keine exakte Lösung zu 9960 | Nächste Lösung siehe 9950

Keine exakte Lösung zu 9970 | Nächste Lösung siehe 9950

Keine exakte Lösung zu 9980 | Nächste Lösung siehe 9990

Links: 9990 + [5] | Rechts: [21, 21, 29, 259, 287, 399, 2993, 2993, 2993]

Links: 10000 + [] | Rechts: [5, 21, 21, 29, 259, 287, 399, 2993, 2993, 2993]

Abdeckung: 61.0% - 610 Kombinationen

Benötigte Zeit: 0.137s

Beispiel 5

Summen gekürzt: 121779

Summen ungekürzt: 8388608

Keine exakte Lösung zu 10 | Nächste Lösung siehe 20

Links: 20 + [11, 99480] | Rechts: [99511]

Keine exakte Lösung zu 30 | Nächste Lösung siehe 20

Keine exakte Lösung zu 40 | Nächste Lösung siehe 20

Keine exakte Lösung zu 50 | Nächste Lösung siehe 20

Keine exakte Lösung zu 60 | Nächste Lösung siehe 20

Keine exakte Lösung zu 70 | Nächste Lösung siehe 100

Keine exakte Lösung zu 80 | Nächste Lösung siehe 100

Keine exakte Lösung zu 90 | Nächste Lösung siehe 100

Links: 100 + [299836, 299836] | Rechts: [11, 599761]

Keine exakte Lösung zu 110 | Nächste Lösung siehe 100

Links: 120 + [99480, 299836, 299836] | Rechts: [99511, 599761]

Keine exakte Lösung zu 130 | Nächste Lösung siehe 120

Keine exakte Lösung zu 140 | Nächste Lösung siehe 150

Links: 150 + [99511, 599761, 94499810, 661499326, 661499326, 661499326] | Rechts: [11, 99480, 299836, 299836, 4497786, 4497786, 1499171, 10499654, 10499654, 10499654, 10499654, 41999427, 1984505261]

Keine exakte Lösung zu 160 | Nächste Lösung siehe 150

Links: 170 + [599761, 94499810, 661499326, 661499326, 661499326] | Rechts: [299836, 299836, 4497786, 4497786, 1499171, 10499654, 10499654, 10499654, 10499654, 41999427, 1984505261]

Keine exakte Lösung zu 180 | Nächste Lösung siehe 170

Links: 190 + [11, 99480, 599761, 94499810, 661499326, 661499326, 661499326] | Rechts: [99511, 299836, 299836, 4497786, 4497786, 1499171, 10499654, 10499654, 10499654, 10499654, 41999427, 1984505261]

Keine exakte Lösung zu 200 | Nächste Lösung siehe 190

Keine exakte Lösung zu 210 | Nächste Lösung siehe 190

Keine exakte Lösung zu 220 | Nächste Lösung siehe 190

Keine exakte Lösung zu 230 | Nächste Lösung siehe 190

Keine exakte Lösung zu 240 | Nächste Lösung siehe 270

Keine exakte Lösung zu 250 | Nächste Lösung siehe 270

Keine exakte Lösung zu 260 | Nächste Lösung siehe 270

Links: 270 + [94499810, 661499326, 661499326, 661499326] | Rechts: [11, 4497786, 4497786, 1499171, 10499654, 10499654, 10499654, 41999427, 1984505261]

Keine exakte Lösung zu 280 | Nächste Lösung siehe 270

Links: 290 + [99480, 94499810, 661499326, 661499326, 661499326] | Rechts: [99511, 4497786, 4497786, 1499171, 10499654, 10499654, 10499654, 10499654, 41999427, 1984505261]

Keine exakte Lösung zu 300 | Nächste Lösung siehe 290

Keine exakte Lösung zu 310 | Nächste Lösung siehe 290

Keine exakte Lösung zu 320 | Nächste Lösung siehe 290

Keine exakte Lösung zu 330 | Nächste Lösung siehe 290

Keine exakte Lösung zu 340 | Nächste Lösung siehe 290

Keine exakte Lösung zu 350 | Nächste Lösung siehe 390

Keine exakte Lösung zu 360 | Nächste Lösung siehe 390

Keine exakte Lösung zu 370 | Nächste Lösung siehe 390

Keine exakte Lösung zu 380 | Nächste Lösung siehe 390

Links: 390 + [99480, 299836, 299836, 94499810, 661499326, 661499326, 661499326] | Rechts: [11, 99511, 599761, 4497786, 4497786, 1499171, 10499654, 10499654, 10499654, 10499654, 41999427, 1984505261]

Keine exakte Lösung zu 400 | Nächste Lösung siehe 390

Keine exakte Lösung zu 410 | Nächste Lösung siehe 390

Keine exakte Lösung zu 420 | Nächste Lösung siehe 390

Keine exakte Lösung zu 430 | Nächste Lösung siehe 390

Keine exakte Lösung zu 440 | Nächste Lösung siehe 390

Keine exakte Lösung zu 450 | Nächste Lösung siehe 390

Keine exakte Lösung zu 460 | Nächste Lösung siehe 390

Keine exakte Lösung zu 470 | Nächste Lösung siehe 390

Keine exakte Lösung zu 480 | Nächste Lösung siehe 390

Keine exakte Lösung zu 490 | Nächste Lösung siehe 390

Keine exakte Lösung zu 500 | Nächste Lösung siehe 390

Keine exakte Lösung zu 9500 | Nächste Lösung siehe 9530

Keine exakte Lösung zu 9510 | Nächste Lösung siehe 9530

Keine exakte Lösung zu 9520 | Nächste Lösung siehe 9530

Links: 9530 + [99511, 299836, 299836, 4497786, 4497786, 1499171, 10499654, 10499654, 10499654, 10499654, 41999427, 94499810, 94499810] | Rechts: [11, 99480, 599761, 283501867]

Keine exakte Lösung zu 9540 | Nächste Lösung siehe 9530

Links: 9550 + [299836, 299836, 4497786, 4497786, 1499171, 10499654, 10499654, 10499654, 10499654, 41999427, 94499810, 94499810] | Rechts: [599761, 283501867]

Keine exakte Lösung zu 9560 | Nächste Lösung siehe 9550

Links: 9570 + [11, 99480, 299836, 299836, 4497786, 4497786, 1499171, 10499654, 10499654, 10499654, 10499654, 41999427, 94499810, 94499810] | Rechts: [99511, 599761, 283501867]

Keine exakte Lösung zu 9580 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9590 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9600 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9610 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9620 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9630 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9640 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9650 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9660 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9670 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9680 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9690 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9700 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9710 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9720 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9730 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9740 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9750 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9760 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9770 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9780 | Nächste Lösung siehe 9570

Keine exakte Lösung zu 9790 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9800 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9810 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9820 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9830 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9840 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9850 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9860 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9870 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9880 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9890 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9900 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9910 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9920 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9930 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9940 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9950 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9960 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9970 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9980 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 9990 | Nächste Lösung siehe 9570
Keine exakte Lösung zu 10000 | Nächste Lösung siehe 9570
Abdeckung: 5.7% - 57 Kombinationen
Benötigte Zeit: 21.644s

Quellcode

```
/**
 * Methode zur Berechnung der Potenzmenge
 * @param originalSet Set für das die Submengen berechnet werden sollen
 * @return Alle Submengen für die zuvor angegebene Menge
 */
public static LinkedList<LinkedList<Integer>> alleSubmengen(List<Integer> originalSet) {
    LinkedList<LinkedList<Integer>> sets = new LinkedList<>();
    if (originalSet.isEmpty()) {
        sets.add(new LinkedList<>());
        return sets;
    }
    Integer head = originalSet.get(0);
    for (LinkedList<Integer> set : alleSubmengen(originalSet.subList(1, originalSet.size()))) {
        LinkedList<Integer> newSet = new LinkedList<>();
        newSet.add(head);
        newSet.addAll(set);
        System.out.println("Added " + newSet);
        System.out.println("Added set " + set);
        sets.add(newSet);
        sets.add(set);
    }
}
```

```

    return sets;
}

/**
 * Berechnet anhand der Gewichte alle möglichen darstellbaren Gewichte
 */
public void generateSolutions() {
    // Berechnung aller möglichen Submengen und deren
    HashMap<Integer, LinkedList<Integer>> subMengen = new HashMap<>();
    for (LinkedList<Integer> subSet : alleSubmengen(weights)) {
        Integer sum = sum(subSet);
        if (sum > 0)
            subMengen.put(sum, subSet);
    }
    // Iteration über alle darzustellenden Gewichte
    for (int darzustellendesGew = 10; darzustellendesGew <= 10000; darzustellendesGew += 10) {
        List<Integer> gewichteLinks = new ArrayList<>();
        List<Integer> gewichteRechts = new ArrayList<>();

        boolean solutionFound = false;

        if (subMengen.containsKey(darzustellendesGew)) {
            gewichteRechts.addAll(subMengen.get(darzustellendesGew));
            solutionFound = true;
        }

        if (!solutionFound) { // Falls sich das darzustellende Gewicht nicht durch eine einfache Summe darstellen lässt
            // Es wird sich jede Submenge betrachtet und untersucht,
            // ob deren Summe plus das darzustellende Gewicht, der Summe einer anderen Submenge entspricht.
            for (Map.Entry<Integer, LinkedList<Integer>> integerLinkedListEntry : subMengen.entrySet()) {
                LinkedList<Integer> mengeAnGewichten = integerLinkedListEntry.getValue();
                Integer summeDerGewichte = integerLinkedListEntry.getKey();
                int summeLinks = summeDerGewichte + darzustellendesGew;
                if (subMengen.containsKey(summeLinks)) {
                    // Sollte es möglich sein, wird überprüft ob die dabei verwendeten Gewichte
                    // in der Gesamtmenge an Gewichten enthalten sind
                    List<Integer> gewichteKopie = new LinkedList<>(weights);
                    gewichteKopie.removeAll(mengeAnGewichten);
                    LinkedList<Integer> ausgleichsGewichte = subMengen.get(summeLinks);
                    if (gewichteKopie.containsAll(ausgleichsGewichte)) {
                        gewichteLinks.addAll(mengeAnGewichten);
                        gewichteRechts.addAll(ausgleichsGewichte);
                        solutionFound = true;
                        break;
                    }
                }
            }
        }

        // Sollte über einen der beiden Wege eine Lösung gefunden worden sein,
        // wird diese der Menge an Gewichtsdarstellungen hinzugefügt
        if (solutionFound) this.gewichtDarstellungen.add(new Gewichtsdarstellung(gewichteLinks, gewichteRechts,
            darzustellendesGew));
    }
    // Ausgabe der Gewichte
    for (int darzustellendesGew = 10; darzustellendesGew <= 10000; darzustellendesGew += 10) {
        Gewichtsdarstellung nearest = null;
        boolean exakteLoesung = false;
        for (Gewichtsdarstellung gewichtsDarstellung : gewichtDarstellungen) {
            if (gewichtsDarstellung.getDarzustellendesGewicht() == darzustellendesGew) {
                System.out.println(gewichtsDarstellung.toString());
                exakteLoesung = true;
                break;
            }
        }
    }
}

```

```

    }
}
if (!exakteLoesung) {
    for (Gewichtsdarstellung gewichtsdarstellung : gewichtDarstellungen) {
        if (nearest == null ||
            Math.abs(gewichtsdarstellung.getDarzustellendesGewicht() - darzustellendesGew)
                < Math.abs(nearest.getDarzustellendesGewicht() - darzustellendesGew)) {
            nearest = gewichtsdarstellung;
        }
    }
    System.out.println("Keine exakte Lösung zu " + darzustellendesGew + " | Nächste Lösung siehe " +
nearest.getDarzustellendesGewicht());
}
}
System.out.println("Coverage: " + (gewichtDarstellungen.size() / 1000.0) * 100 + "% - " +
gewichtDarstellungen.size() + " Combinations");
}

```