RUCKSÄCKE EFFIZIENT PACKEN

VERSUCH DER ENTWICKLUNG EINES APPROXIMATIONSALGORITHMUS FÜR DAS *SIMPLE KNAPSACK PROBLEM (SKP) IM 9./10. SCHULJAHR*

WAS IST DAS „EINFACHE RUCKSACKPROBLEM“ (SKP)?

EINE (OPTIONALE) GESCHICHTE ZUM EINSTIEG

Als Einstieg in die Unterrichtseinheit kann folgende Geschichte[1] verwendet werden. Sie soll einerseits die Lernenden dazu anstossen, sich mit der Frage, was die wirklich wichtigen Dinge im Leben sind, zu beschäftigen, andererseits beleuchtet sie den Umstand, dass die Grösse einzelner Elemente, die man in ein Gefäss gibt, massgeblich dafür ist, wie gut das Gefäss am Ende gefüllt werden kann.

Weil diese Geschichte allerdings nur sehr am Rande mit dem Thema zu tun hat, ist dieser Einstieg keineswegs essenziell, sondern – wie im Titel schon angegeben – lediglich optional.

Ein Professor stand vor seiner Philosophie-Klasse und hatte einige Gegenstände vor sich. Als der Unterricht begann, nahm er wortlos einen Rucksack und begann, diesen mit Golfbällen zu füllen. Er fragte die Studierenden, ob der Rucksack nun voll sei. Sie bejahten es. Dann nahm der Professor einen Becher mit Kieselsteinen und schüttete diese in den Rucksack.

Er bewegte den Rucksack sachte und die Kieselsteine rollten in die Leerräume zwischen den Golfbällen. Dann fragte er die Studierenden wiederum, ob der Rucksack nun voll sei. Sie stimmten zu. Der Professor nahm als Nächstes eine Dose mit Sand und schüttete diesen in den Rucksack. Natürlich füllte der Sand den kleinsten verbliebenen Freiraum. Er fragte wiederum, ob der Rucksack nun voll sei. Die Studierenden antworteten einstimmig: „Ja!“

Der Professor holte eine Limonade und eine Tasse Kaffee unter dem Tisch hervor und schüttete den ganzen Inhalt in den Rucksack und füllte somit den letzten Raum zwischen den Sandkörnern aus. Die Studierenden lachten. „Nun,“, sagte der Professor, als das Lachen langsam nachliess, „ich möchte, dass Sie diesen Rucksack als ein Bild für Ihr Leben betrachten. Die Golfbälle sind die wichtigen Dinge in Ihrem Leben: Ihre Familie, Ihre Kinder, Ihre Gesundheit, Ihre Freunde, die bevorzugten, ja leidenschaftlichen Aspekte ihres Lebens, mit denen, falls in Ihrem Leben alles verloren ginge und nur noch diese verbleiben würden, ihr Leben trotzdem noch erfüllend wäre.

Die Kieselsteine symbolisieren die weiteren Dinge im Leben wie Ihre Arbeit, Ihr Haus, Ihr Auto. Der Sand ist alles andere, die Kleinigkeiten. Falls Sie den Sand zuerst in den Rucksack geben“, fuhr der Professor fort, „hat es weder Platz für die Kieselsteine noch für die Golfbälle. Dasselbe gilt für Ihr Leben. Wenn Sie all Ihre Zeit und Energie in Kleinigkeiten investieren, werden Sie nie Platz haben für die wichtigen Dinge. Achten Sie auf die Dinge, welche Ihr Glück gefährden. Spielen Sie mit den Kindern. Nehmen Sie sich Zeit für eine medizinische Untersuchung. Führen Sie Ihren Partner zum Essen aus. Es wird immer noch Zeit bleiben, um Pflichten zu erledigen. Achten Sie zuerst auf die Golfbälle, die Dinge, die wirklich wichtig sind. Setzen Sie Ihre Prioritäten. Der Rest ist nur Sand.“

Eine Studentin erhob die Hand und wollte wissen, was der Kaffee und die Limonade bedeuten sollen. Der Professor schmunzelte: „Ich bin froh, dass Sie das fragen. Sie sind dafür da, Ihnen zu zeigen, dass, egal wie schwierig Ihr Leben auch sein mag, es immer noch Platz hat für eine Tasse Kaffee oder ein Getränk mit Ihren besten Freundinnen und Freunden.“

[1] Als Lehrperson für „Ethik und Religionen“ kann ich auf einen reichen Fundus an Geschichten zurückgreifen, die sich damit beschäftigen, wie das Leben gelingen kann. An diese Geschichte musste ich unweigerlich denken, als wir uns in der Vorlesung „Algorithmen und Datenstrukturen II“ bei Professor Dennis Komm mit dem *Simple Knapsack Problem (SKP)* beschäftigten. Leider ist mir der Urheber dieser Geschichte unbekannt, ausserdem habe ich die Geschichte gegenüber dem Original leicht abgeändert.

ERKLÄRUNG DES EINFACHEN RUCKSACKPROBLEMS

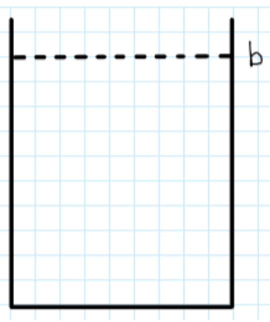


Abbildung 1 - ein leerer "Rucksack"

Als Erstes soll die Frage beantwortet werden: Worum geht's eigentlich? Dabei soll die Lehrperson das Problem möglichst anschaulich erklären, aber nicht auf eine saubere mathematische Schreibweise verzichten.

Wir haben ein Gefäss, und zwar einen „Rucksack". In diesen Rucksack möchten wir viele Dinge einpacken, aber der Rucksack hat eine bestimmte **Füllgrenze b** $\in \mathbb{N}$. Eine Eingabe für das SKP besteht aus diesem b und den einzelnen Objekten, genauer gesagt: den **Gewichten** $w_1, w_2, \ldots, w_n \in \mathbb{N}$ aller Objekte, die wir in den Rucksack packen könnten. (Man spricht von „Gewichten", auch wenn in der Vorstellung bei einem Rucksack das Volumen der Dinge die anschaulichere Grösse wäre.)

Man schreibt: $\alpha = (w_1, w_2, \ldots, w_n, b)$

Unsere **Aufgabe** ist es nun, ausgewählte Objekte so in den Rucksack zu packen, dass die Füllgrenze dabei nicht überschritten wird. Alle möglichen Kombinationen von Gewichten, welche diese Bedingung erfüllen, sind **zulässige Lösungen**.

Unser **Ziel** ist es, aus allen zulässigen Lösungen die **optimale Lösung** zu finden, nämlich jene, bei der die Summe der Gewichte der eingepackten Objekte im Rucksack am grössten ist.

Ein Beispiel: $\alpha = (1, 2, 3, 4, 2, 10)$

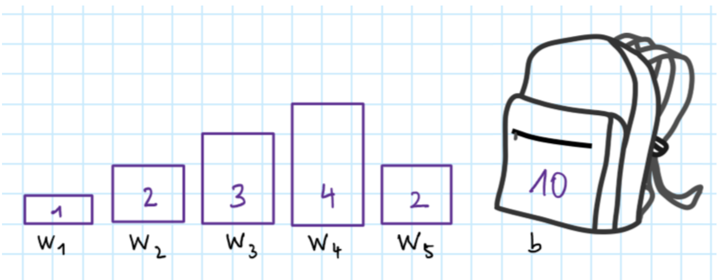


Abbildung 2 – eine einfache Instanz des SKP

Aufgabe 1: *Bestimmt, welche der folgenden Lösungskandidaten für das angegebene Beispiel zulässige Lösungen sind!*

a) w_1, w_2, w_3, w_4, w_5
b) w_1, w_3, w_5
c) w_2, w_4, w_5
d) w_4, w_2, w_3
e) w_3
f) w_2, w_3, w_3, w_4

Zulässige Lösungen sind alle, ausgenommen a), weil hier die Füllgrenze des Rucksacks überschritten wird, und f), weil hier ein Objekt (w_3) zweimal in den Rucksack gepackt wird.

Aufgabe 2: *Findet für das angegebene Beispiel aus allen zulässigen Lösungen die optimale Lösung!*

Die optimale Lösung ist nicht in den Vorschlägen a) bis f) zu finden. Tatsächlich gibt es zwei optimale Lösungen, nämlich w_1, w_2, w_3, w_4 und w_1, w_3, w_4, w_5.

Probleme, bei denen es darum geht, eine optimale Lösung zu finden, nennt man „**Optimierungsprobleme**". Es handelt sich beim SKP um ein Optimierungsproblem, weil wir den Rucksack so voll wie möglich machen wollen.

- <u>Eingabe</u>: ein Tupel α positiver ganzer Zahlen für die Gewichte von n Objekten und die Rucksackkapazität b
- <u>Zulässige Lösungen</u>: Alle Teilmengen T der Menge der Indizes der Gewichte, deren Summe („Gesamtgewicht") kleiner oder gleich der Rucksackkapazität ist:
  $T \subseteq \{1, 2, \dots, n\}$, wobei $(\text{Summe der Gewichte aller Elemente von T}) \leq b$
- <u>Kosten</u>: $\text{cost}(T) = \text{Summe der Gewichte aller Elemente von T}$
- <u>Ziel</u>: $\max$

Eine zentrale Frage für die Lösung des Problems ist, wie viele Teilmengen T es insgesamt gibt. Auf die Beantwortung dieser Frage wird an dieser Stelle vorerst bewusst verzichtet, wohl aber soll die Lehrperson auf die Wichtigkeit dieser Frage hinweisen und ankündigen, dass später noch ausführlich auf die Frage der Komplexität des Problems eingegangen wird.

Dennoch sollen für das angegebene Beispiel plakativ einmal **alle zulässigen Lösungen** aufgelistet werden, um zu verdeutlichen, dass auch alle Teilmengen, die weitestmöglich von der optimalen Lösung entfernt sind, als Lösungen zulässig sind:

$T_0 = \{\}$	$T_6 = \{1, 2\}$	$T_{12} = \{2, 5\}$	$T_{18} = \{1, 2, 5\}$	$T_{24} = \{2, 4, 5\}$
$T_1 = \{1\}$	$T_7 = \{1, 3\}$	$T_{13} = \{3, 4\}$	$T_{19} = \{1, 3, 4\}$	$T_{25} = \{3, 4, 5\}$
$T_2 = \{2\}$	$T_8 = \{1, 4\}$	$T_{14} = \{3, 5\}$	$T_{20} = \{1, 3, 5\}$	$T_{26} = \{1, 2, 3, 4\}$
$T_3 = \{3\}$	$T_9 = \{1, 5\}$	$T_{15} = \{4, 5\}$	$T_{21} = \{1, 4, 5\}$	$T_{27} = \{1, 2, 3, 5\}$
$T_4 = \{4\}$	$T_{10} = \{2, 3\}$	$T_{16} = \{1, 2, 3\}$	$T_{22} = \{2, 3, 4\}$	$T_{28} = \{1, 2, 4, 5\}$
$T_5 = \{5\}$	$T_{11} = \{2, 4\}$	$T_{17} = \{1, 2, 4\}$	$T_{23} = \{2, 3, 5\}$	$T_{29} = \{1, 3, 4, 5\}$

Die optimalen Lösungen sind $T_{26} = \{1, 2, 3, 4\}$, bestehend aus den Gewichten w_1, w_2, w_3, w_4, und $T_{29} = \{1, 3, 4, 5\}$, bestehend aus den Gewichten w_1, w_3, w_4, w_5. Die Kosten sind beide Male identisch:

$$\mathbf{cost(T_{26})} = w_1 + w_2 + w_3 + w_4 = 1 + 2 + 3 + 4 = \underline{\mathbf{10}}$$

$$\mathbf{cost(T_{29})} = w_1 + w_3 + w_4 + w_5 = 1 + 3 + 4 + 2 = \underline{\mathbf{10}}$$

<u>Anmerkung:</u>
Dass ein konkretes Gewicht mehrmals in der Eingabe vorkommt, das heisst, dass **mehrere Objekte das gleiche Gewicht** haben, ist möglich. Die einzelnen Objekte müssen also nicht zwangsläufig unterschiedliche Gewichte haben. Im angegebenen Beispiel kommt bei den Gewichten der Wert 2 zweimal vor, nämlich in den Gewichten w_2 und w_5. Weil der Wert 2 in der optimalen Lösung vorkommt, gibt es zwei unterschiedliche optimale Lösungen, die jedoch die gleichen Gewichte beinhalten.

EINFACHE BEISPIELE ZUM KENNENLERNEN DES PROBLEMS UND ZUM NACHDENKEN ÜBER LÖSUNGSSTRATEGIEN

Der Übersichtlichkeit halber werden wir im weiteren Verlauf die Lösungen nicht als Mengen T der Indizes, sondern als Folgen von Gewichten $w_a, w_b, w_c \ldots$ anschreiben, beim angegebenen Beispiel für die Lösung $T_{12} = \{2, 5\}$ etwa in der Form $(2, 2)$ oder für die Lösung $T_{25} = \{3, 4, 5\}$ etwa in der Form $(1, 3, 4)$ – wobei die Reihenfolge der Gewichte in diesen Folgen nicht von Belang ist.

Wenn bei verschiedenen Teilmengen in den Folgen der Gewichte die gleichen konkreten Gewichte vorkommen, so soll es bei den folgenden Aufgaben ausreichen, wenn diese Folgen nur ein einziges Mal angeschrieben werden, auch wenn damit mehrere verschiedene zulässige Lösungen in einer einzigen Folge zusammengefasst werden. Im angegebenen Beispiel würde die Folge $(2, 3)$ nur einmal angeschrieben, wohl wissend, dass damit die unterschiedlichen Lösungen $T_{10} = \{2, 3\}$ und $T_{14} = \{3, 5\}$ zusammengefasst werden.

Aufgabe 3: *Findet zu folgenden Instanzen des einfachen Rucksackproblems eine optimale Lösung!*

a) $\alpha = (4, 8, 2, 6, 8, 2, 2, 1, 17)$
b) $\beta = (2, 7, 4, 3, 1, 5, 3, 15)$
c) $\gamma = (6, 8, 2, 4, 3, 1, 11, 9, 5, 23)$

Das Finden einer optimalen Lösung (alle Möglichkeiten sind in Aufgabe 4 aufgelistet) sollte für die SuS einfach sein. Es geht hauptsächlich darum, sich an die Notation zu gewöhnen. Für die Entwicklung erster Lösungsstrategien sollte mit Aufgabe 4 ein erster Schritt getan werden.

Aufgabe 4: *Findet zu den Instanzen aus Aufgabe 3 sämtliche optimale Lösungen! Was wäre eine gute Strategie, um sicherzugehen, dass wirklich sämtliche optimale Lösungen (und dies möglichst schnell) gefunden werden?*

Lösungen:

a) $(8, 8, 1), (8, 6, 2, 1), (8, 4, 2, 2, 1), (6, 4, 2, 2, 2, 1)$
b) $(7, 5, 3), (7, 5, 2, 1), (7, 4, 4), (7, 4, 3, 1), (7, 3, 3, 2), (5, 4, 3, 3), (5, 4, 3, 2, 1)$
c) $(11, 9, 3), (11, 9, 2, 1), (11, 8, 4), (11, 8, 3, 1), (11, 6, 5, 1), (11, 6, 4, 2), (11, 5, 4, 3), (11, 5, 4, 2, 1), (9, 8, 6), (9, 8, 5, 1), (9, 8, 4, 2), (9, 8, 3, 2, 1), (9, 6, 5, 2, 1), (9, 6, 4, 3, 1), (9, 5, 4, 3, 2), (8, 6, 5, 4), (8, 6, 5, 3, 1), (8, 6, 4, 3, 2), (8, 5, 4, 3, 2, 1)$

Die Lösung zu Aufgabe 4 a ist einfach zu finden, diejenige zu Aufgabe 4 b dürfte ebenfalls keine grösseren Probleme bereiten. **Nach** dem gemeinsamen Überprüfen von Aufgabe 4 b und **vor** dem Lösen von Aufgabe 4 c ist ein **Gespräch über die von den SuS verwendeten Strategien** förderlich.

Es ist davon auszugehen, dass in diesem Gespräch als eine mögliche Strategie genannt wird, dass man bei der Suche nach allen Lösungen jeweils mit dem grössten Gewicht beginnt und die weiteren Gewichte in abnehmender Grösse dazu kombiniert, solange die Füllgrenze des Rucksacks noch nicht erreicht ist. Alternativ kann die Lehrperson vorschlagen, diese Strategie in Betracht zu ziehen. Die Erkenntnis, die festgehalten werden soll, ist, dass es für gewisse Strategien hilfreich sein könnte, wenn man **die Gewichte der Grösse nach sortiert**.

ETWAS SCHWIERIGERE BEISPIELE ZUM NACHDENKEN ÜBER DIE KOMPLEXITÄT DES PROBLEMS UND MÖGLICHKEITEN ZUR REDUKTION DER KOMPLEXITÄT

Aufgabe 5: *Findet zu folgenden nicht mehr ganz so einfachen Instanzen des SKP eine möglichst gute Lösung! Ihr dürft dazu den Taschenrechner benutzen, habt aber pro Instanz nur maximal zwei Minuten Zeit!*

a) $\delta = (329, 302, 299, 305, 612, 258, 315, 287, 306, 300, 245, 1247)$
b) $\varepsilon = (308, 89, 448, 122, 188, 821, 248, 598, 258, 165, 103, 279, 555, 22, 55, 765, 76, 234, 255, 898, 14, 3894)$

Die Zeit wird jeweils von der Lehrperson gestoppt. Ziel dieser Aufgabe ist es nicht, eine optimale Lösung zu finden (auch wenn das für begabte SuS bei Aufgabe 3 a auch in nur zwei Minuten möglich ist), sondern das Ziel ist es, dass die SuS erkennen, wie komplex das Problem wird, wenn die Anzahl der Gewichte zunimmt.

Gleichwohl ist es nach dieser Aufgabe sinnvoll, die Qualität der gefundenen Lösungen zu vergleichen, weshalb hier die optimalen Lösungen genannt seien: Für Aufgabe 5 a ist es $(612, 329, 306)$, und für Aufgabe 5 b ist es $(898, 821, 765, 598, 555, 188, 55, 14)$.

Im anschliessenden **Gespräch** soll wieder auf die Frage der Strategie eingegangen werden. Es liegt im Bereich des Möglichen, dass es SuS gibt, die hier schon intuitiv die **Greedy-Methode** angewendet haben. Falls nicht, soll an dieser Stelle auf diese Möglichkeit hingewiesen werden.

ALGORITHMUS GREEDY-SKP

Eingabe: $w_1, w_2, \dots, w_n, b \in \mathbb{N}\setminus\{0\}$

1. **Sortiere $w_1, w_2, \dots, w_n$ absteigend.**
   Falls es Gewichte gibt, die grösser als b sind, werden sie bei der Sortierung entfernt.
   Somit gilt: $b \geq w_1 \geq w_2 \geq \dots \geq w_n$
2. **Initialisierung:** $T = \{\}$ $cost(T) = 0$
   Wir beginnen also mit einem leeren Rucksack.
3. **Lösungsfindung**: Gehe alle Gewichte der Reihe nach durch und nimm jene Gewichte in den Rucksack auf, durch deren Hinzufügung die Füllgrenze b des Rucksacks nicht überschritten wird.

Ausgabe: **T**

Beispiel:

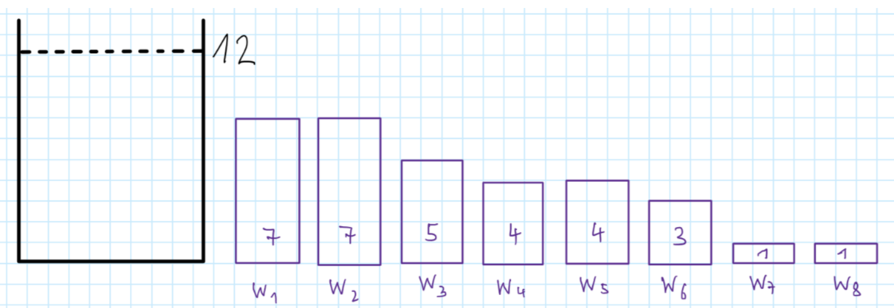


Abbildung 3 – ein leerer Rucksack mit der Grösse nach sortierten Gewichten

Der Algorithmus würde nun folgendermassen vorgehen:

$$T = \{\} \quad cost(T) = 0 \quad \rightarrow \quad T = \{1\} \quad cost(T) = 7 \quad \rightarrow \quad T = \{1, 3\} \quad cost(T) = 12$$

ZUR KOMPLEXITÄT DES „EINFACHEN RUCKSACKPROBLEMS" (SKP)

Es handelt sich beim SKP um ein sogenanntes **NP-schweres Problem**. Das bedeutet, salopp formuliert: Um eine optimale Lösung zu finden, kennen wir keine effizientere Strategie als alle möglichen Kombinationen von Gewichten (= alle Teilmengen T_i) durchzuprobieren, bis wir sicher die optimale Lösung gefunden haben. Die Anzahl der möglichen Kombinationen nimmt mit der Länge der Eingabe exponentiell zu (2^n).

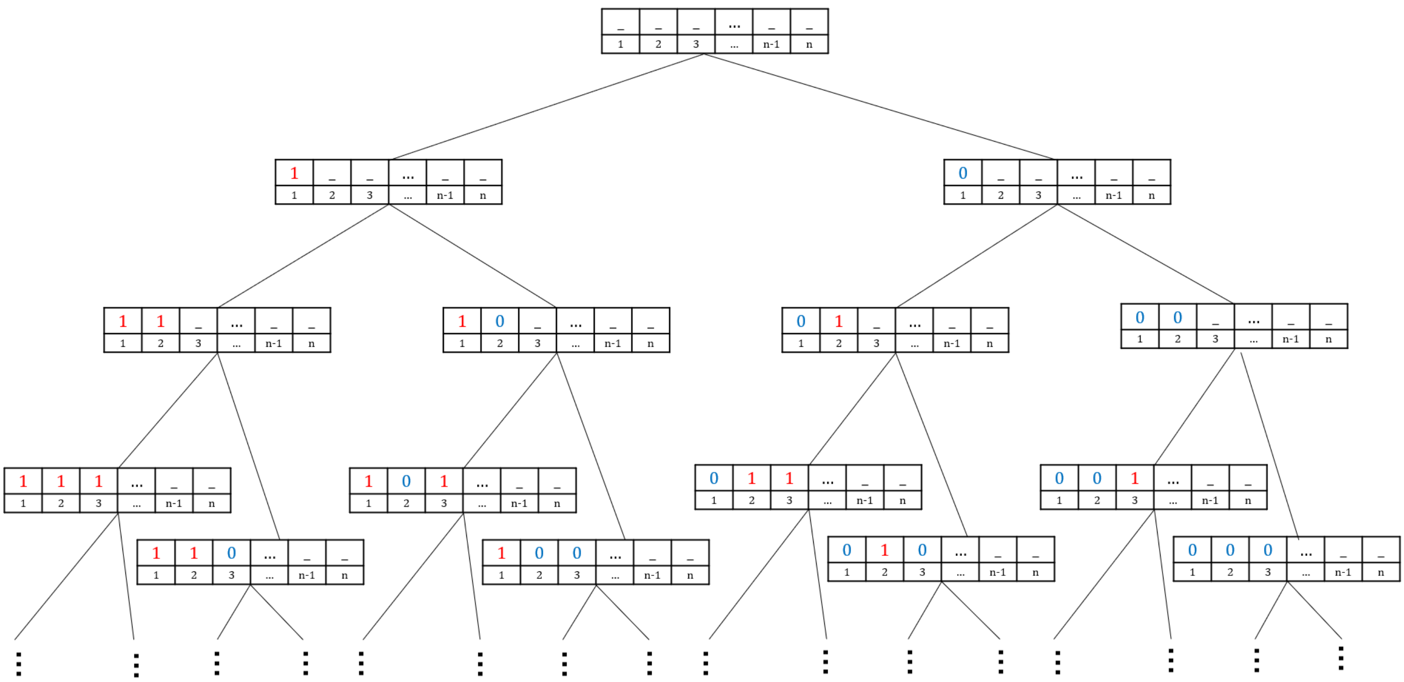
Um dies für die SuS zu veranschaulichen, soll an dieser Stelle auf die bereits bei der Erklärung des Problems angekündigte Frage eingegangen werden, **wie viele Teilmengen T_i es bei einer bestimmten Anzahl n an Gewichten gibt**, das heisst, wie komplex die Aufgabe für einen Computer wird, falls er nicht die Greedy-Methode anwendet, sondern <u>alle</u> Teilmengen überprüft, um mit Sicherheit ein optimales Ergebnis zu liefern.

Aufgabe 6: *Findet <u>alle</u> Teilmengen T_i der folgenden Mengen M_i, welche die Fälle repräsentieren, bei denen die Eingabe für das SKP aus 1, 2 oder 3 Gewichten besteht! Lässt sich ein Muster erkennen?*

a) $M_1 = \{1\}$ b) $M_2 = \{1, 2\}$ c) $M_3 = \{1, 2, 3\}$

Tatsächlich ist die **Anzahl der Teilmengen T** der Menge der Indizes von n Gewichten $\mathbf{2^n}$.

Dass dem so ist, lässt sich verhältnismässig einfach und anschaulich erklären, etwa mithilfe eines Baumdiagramms. Weil jedes der n Objekte entweder in einer Teilmenge vorhanden oder nicht vorhanden sein kann, können die möglichen Teilmengen als eine Binärfolge der Länge n dargestellt werden, wobei 0 bedeutet, dass das Objekt nicht in der Teilmenge ist, und 1, dass sich das Objekt in der Teilmenge befindet.



Letzten Endes wird, wenn an allen n Stellen Nullen und Einsen eingesetzt werden, die **Anzahl der Blätter dieses Binärbaumes 2^n** sein.

Alternativ oder ergänzend kann auch mit einer Tabelle gezeigt werden, dass sich mit jedem zusätzlichen Gewicht die Anzahl aller bisherigen Teilmengen verdoppelt. Das ist deshalb so, weil alle bisherigen Teilmengen natürlich nach wie vor Teilmengen bleiben und man allen bisherigen Teilmengen (auch der leeren Menge) das zusätzliche Gewicht hinzufügen kann, wodurch man eine genauso grosse Anzahl an neuen Teilmengen erhält.

Das kann an den kleinsten Mengen an Gewichten anhand eines sich schrittweise aufbauenden **Tafelbildes** veranschaulicht werden:

Anzahl Gewichte	Teilmengen	Anzahl Teilmengen
0	$\{\}$	1
1	$\{\}, \{w_1\}$	2
2	$\{\}, \{w_1\}, \{w_2\}, \{w_1, w_2\}$	4
3	$\{\}, \{w_1\}, \{w_2\}, \{w_1, w_2\}, \{w_3\}, \{w_1, w_3\}, \{w_2, w_3\}, \{w_1, w_2, w_3\}$	8
4	$\{\}, \{w_1\}, \{w_2\}, \{w_1, w_2\}, \{w_3\}, \{w_1, w_3\}, \{w_2, w_3\}, \{w_1, w_2, w_3\}$, $\{w_4\}, \{w_1, w_4\}, \{w_2, w_4\}, \{w_1, w_2, w_4\}, \{w_3, w_4\}, \{w_1, w_3, w_4\}$, $\{w_2, w_3, w_4\}, \{w_1, w_2, w_3, w_4\}$	16

Vermutlich bereits in der vierten, spätestens aber in der fünften Zeile der Tabelle sollten die SuS erkennen, dass die Anzahl der Teilmengen T bei einer Anzahl n an Gewichten genau 2^{n} ist.

Um zu veranschaulichen, wie „schrecklich" eine exponentielle Laufzeit ist, kann noch mithilfe eines **Rechenprogramms** gezeigt werden, wie sich die Anzahl der Teilmengen entwickelt, wenn die Anzahl der Gewichte 20, 50 oder 100 ist.

Weil jedes der n Gewichte gemäss den obigen Darstellungen in der Hälfte aller Teilmengen vorkommt, ist die durchschnittliche Grösse aller Teilmengen $\frac{n}{2}$. Somit kann der Aufwand für den Computer (der damit bestimmt werden soll, wie viele einzelne Gewichte angesehen werden müssen) angegeben werden mit $\frac{n}{2} \cdot 2^n = n \cdot 2^{n-1}$.

Diesem Aufwand wird nun der Aufwand gegenübergestellt, der sich für einen Computer stellt, wenn der Algorithmus **Greedy-SKP** angewendet wird.

Der Aufwand für das Sortieren der n Gewichte ist $n \cdot \log_2 n$.[2] Und die Anzahl der einzelnen Gewichte, die angesehen werden müssen, ist genau n, weil der Computer die Liste der Gewichte genau ein einziges Mal durchgehen muss. Somit kann der Aufwand angegeben werden mit $n \cdot \log_2 n + n$.

Auch hier kann noch mithilfe eines **Rechenprogramms** gezeigt werden, wie sich der Aufwand entwickelt, wenn die Anzahl der Gewichte 20, 50 oder 100 ist, und mit dem Aufwand für das Finden einer optimalen Lösung verglichen werden. Die Erkenntnis, die festgehalten werden soll, ist, dass **mit der Greedy-Methode der Aufwand für einen Computer (beinahe) vernachlässigbar klein** wird.

[2] Das wird nicht eigens gezeigt, da Sortieralgorithmen und deren Komplexität zum Vorwissen der SuS gehören.

BEISPIELE ZUM NACHDENKEN DARÜBER, WIE GUT DIE GREEDY-METHODE IST UND WIE SIE VERBESSERT WERDEN KÖNNTE

DER BEGRIFF DER „GÜTE" EINES APPROXIMATIONSALGORITHMUS

Aufgabe 7: _Diskutiert, bei welchen Lösungen ein fiktiver Algorithmus „Fiktiv-SKP" im Vergleich mit der optimalen Lösung am besten abschneidet! Begründet eure Antwort!_

a) $\text{Fiktiv-SKP: cost}(T_{\text{Fikt}}) = 3$ $\quad$ $\text{optimal: cost}(T_{\text{Opt}}) = 4$
b) $\text{Fiktiv-SKP: cost}(T_{\text{Fikt}}) = 20$ $\quad$ $\text{optimal: cost}(T_{\text{Opt}}) = 25$
c) $\text{Fiktiv-SKP: cost}(T_{\text{Fikt}}) = 200$ $\quad$ $\text{optimal: cost}(T_{\text{Opt}}) = 400$
d) $\text{Fiktiv-SKP: cost}(T_{\text{Fikt}}) = 5000$ $\quad$ $\text{optimal: cost}(T_{\text{Opt}}) = 5500$

Bei einem **Approximationsalgorithmus** wird darauf verzichtet, für ein Optimierungsproblem garantiert eine optimale Lösung zu finden, anstatt dessen wird eine Näherungslösung berechnet. Der Begriff der **Approximationsgüte** gibt an, wie nahe diese Näherungslösung auch im schlechtesten Fall an die optimale Lösung herankommt.

Manche SuS könnten der Fehlvorstellung unterliegen, dass die Differenz zwischen der optimalen Lösung und der Näherungslösung die bestimmende Grösse für die Approximationsgüte wäre und daher der Algorithmus „Fiktiv-SKP" in Aufgabe 7 bei a) am besten und bei d) am schlechtesten abschneiden würde.

Tatsächlich ist die Approximationsgüte $R_A(x)$ eines Algorithmus A auf irgendeiner Eingabe x das Verhältnis der berechneten (Näherungs-)Lösung zur optimalen Lösung $\text{Opt}(x)$. Falls es sich wie beim SKP um ein Maximierungsproblem handelt, gilt:

$$R_A(x) = \frac{\text{cost}(\text{Opt}(x))}{\text{cost}(A(x))}$$

Dementsprechend beträgt die Güte des Algorithmus „Fiktiv-SKP" in Aufgabe 7 bei a) $\frac{4}{3}$, bei b) $\frac{5}{4}$, bei c) schneidet er mit 2 am schlechtesten und bei d) mit $\frac{11}{10}$ am besten ab.

WIE GUT IST DIE GREEDY-METHODE?

Aufgabe 8: _Wendet bei folgenden Instanzen des SKP die Greedy-Methode an! Bei welchen Beispielen führt sie zum optimalen Ergebnis? Bei welchen Beispielen führt sie zu einem Ergebnis, das dem optimalen Ergebnis sehr nahe ist? Bestimmt jeweils die Approximationsgüte der Lösungen, die ihr mit der Greedy-Methode erhaltet!_

a) $\alpha = (19, 17, 15, 13, 11, 9, 7, 5, 2, 22)$ $\quad$ optimal: $(17, 5), (15, 7), (13, 9), \ldots$
b) $\beta = (5, 4, 4, 3, 3, 3, 2, 2, 2, 2, 1, 1, 1, 15)$ $\quad$ optimal: $(5, 4, 4, 2), (5, 4, 3, 3), \ldots$
c) $\gamma = (17, 16, 15, 14, 13, 12, 11, 10, 9, 23)$ $\quad$ optimal: $(14, 9), (13, 10), \ldots$
d) $\delta = (354, 253, 180, 175, 160, 600)$ $\quad$ optimal: $(253, 180, 160)$

Aufgabe 9: _Findet selbst eine oder mehrere Instanzen des SKP, bei dem die Greedy-Methode gegenüber dem optimalen Ergebnis möglichst schlecht abschneidet! In welchem Verhältnis steht die denkbar schlechteste Lösung mit der Greedy-Methode zur optimalen Lösung (welche im schlechtesten Fall mit der Füllgrenze des Rucksacks identisch ist)?_

Aufgabe 8 ist gedacht als Vorbereitung für Aufgabe 9 und muss nicht näher besprochen werden. Ziel ist es, dass die SuS erkennen, dass die Greedy-Methode umso genauer wird, je kleiner die einzelnen Gewichte im Verhältnis zur Füllgrenze sind.

Im Anschluss daran kann auf die **Geschichte vom Beginn dieser Unterrichtseinheit** verwiesen werden, die veranschaulicht, dass die Grösse der Gewichte entscheidend dafür ist, wie gut die Greedy-Methode abschneidet. Wie voll wäre der Rucksack geworden, wenn der Professor ausschliesslich Golfbälle zur Verfügung gehabt hätte, wenn er vielleicht nur Tennisbälle gehabt hätte oder sogar nur Fussbälle … umgekehrt – wenn wir die Flüssigkeiten einmal ausklammern, weil sie keinen Zusammenhang zum SKP haben –, wie voll wäre er geworden, wenn der Professor ausschliesslich Sandkörner zum Befüllen des Rucksacks zur Verfügung gehabt hätte?

DIE DENKBAR SCHLECHTESTE LÖSUNG MIT DER GREEDY-METHODE

Die denkbar schlechteste Lösung mit der Greedy-Methode ergibt sich, wenn das grösste der Gewichte minimal grösser ist als die Hälfte der Füllgrenze und die kleinsten (mindestens zwei) Gewichte genau so gross sind wie die Hälfte der Füllgrenze, wobei die Füllgrenze eine gerade Zahl ist:

$\text{Eingabe: } \left(\frac{b}{2}+1, \frac{b}{2}, \frac{b}{2}, b\right)$ $\text{Optimale Folge: } \left(\frac{b}{2}, \frac{b}{2}\right)$ $\text{Greedy-Methode: } \left(\frac{b}{2}+1\right)$

Ein konkretes Beispiel:

$\text{Eingabe: } (501, 500, 500, 1000)$ $\text{Optimale Folge: } (500, 500)$ $\text{Greedy-Methode: } (501)$

Je grösser gemäss diesem Schema das b wird, umso mehr nähert sich das Verhältnis der optimalen Lösung zur Lösung durch die Greedy-Methode dem Verhältnis 2:1 an. Der **Algorithmus Greedy-SKP** hat also die **Approximationsgüte 2**.

Aufgabe 10: *Bestimmt, was „Approximationsgüte 2“ bedeutet! Zu welchem Anteil wird der Rucksack mit der Greedy-Methode mit Sicherheit gefüllt? Zu welchem Anteil würde er mit Sicherheit gefüllt werden, wenn die Approximationsgüte 1.5 oder sogar 1.25 wäre?*

Als Erkenntnis soll festgehalten werden: Approximationsgüte 2 bedeutet, dass **mit der Greedy-Methode der Rucksack auf jeden Fall mindestens halbvoll** wird. Bei Approximationsgüte 1.5 wäre der Rucksack mindestens zu $\frac{2}{3}$ gefüllt, bei Approximationsgüte 1.25 mindestens zu $\frac{4}{5}$. (Es handelt sich also jeweils um die Umkehrung des Bruches.)

WIE KÖNNTE DIE GREEDY-METHODE VERBESSERT WERDEN?

Die Greedy-Methode besticht durch ihren äusserst geringen Aufwand. Ein bisschen mehr Rechenarbeit könnten wir unseren modernen Computern aber durchaus zumuten, falls sich dadurch die Approximationsgüte unseres Algorithmus deutlich verbessern lässt. Wie wäre es, wenn wir zwar nicht alle, aber doch **so viele Teilmengen durchprobieren, dass der Aufwand für den Computer in einem aus unserer Sicht noch akzeptablen Rahmen bleibt**, und bei jeder dieser Teilmengen den restlichen freien Platz mit der äusserst effizienten Greedy-Methode ausfüllen?

Dazu müssen wir berechnen, wie viele Teilmengen T der Mächtigkeit $k = |T|$ es bei n zur Verfügung stehenden Objekten gibt.[3]

[3] Ausgegangen wird davon, dass die SuS das Berechnen von Kombinationen ohne Wiederholungen („n tief k“) noch nicht gelernt haben.

<table>
<tr><th>k</th><th>Berechnung</th><th>Anzahl Teilmengen</th></tr>
<tr><td>0</td><td>Weil es die leere Menge nur einmal gibt, muss hier nichts berechnet werden. Weil es aber Instanzen gibt, bei denen die Greedy-Methode auch ohne das Durchprobieren diverser Teilmengen den Rucksack zur Gänze füllt und somit zu einer optimalen Lösung führt, darf darauf nicht verzichtet werden.</td><td>1</td></tr>
<tr><td>1</td><td>Auch hier muss noch nichts berechnet werden. Bei n Objekten gibt es genau n Teilmengen mit 1 Element.</td><td>$\frac{n}{1}$</td></tr>
<tr><td>2</td><td>Hier muss jedes Objekt mit jedem Objekt (ausser sich selbst!) kombiniert werden. Weil aber zum Beispiel die Kombinationen w_1w_2 und w_2w_1 dieselbe Teilmenge $T_1 = \{w_1, w_2\}$ ergeben, muss – wie die Tabelle zeigt – die Summe der Kombinationen durch zwei geteilt werden.
<table>
<tr><th></th><th>w_1</th><th>w_2</th><th>w_3</th><th>w_4</th><th>...</th></tr>
<tr><td>w_1</td><td></td><td>w_1w_2</td><td>w_1w_3</td><td>w_1w_4</td><td>...</td></tr>
<tr><td>w_2</td><td>w_2w_1</td><td></td><td>w_2w_3</td><td>w_2w_4</td><td>...</td></tr>
<tr><td>w_3</td><td>w_3w_1</td><td>w_3w_2</td><td></td><td>w_3w_3</td><td>...</td></tr>
<tr><td>w_4</td><td>w_4w_1</td><td>w_4w_2</td><td>w_4w_3</td><td></td><td>...</td></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td></td></tr>
</table></td><td>$\frac{n \cdot (n-1)}{2 \cdot 1}$</td></tr>
<tr><td>3</td><td>Hier und in weiterer Folge müssen wie bei den Teilmengen mit 2 Elementen alle Kombinationen geteilt werden durch alle möglichen Anordnungen, in denen diese Kombinationen auftreten können. Bei drei Elementen sind das $3 \cdot 2 \cdot 1$ mögliche Anordnungen.</td><td>$\frac{n \cdot (n-1) \cdot (n-2)}{3 \cdot 2 \cdot 1}$</td></tr>
<tr><td>4</td><td>Spätestens an dieser Stelle dürfte das Muster für die Berechnung klar sein.</td><td>$\frac{n \cdot (n-1) \cdot (n-2) \cdot (n-3)}{4 \cdot 3 \cdot 2 \cdot 1}$</td></tr>
</table>

Als Diskussionsgrundlage für die Klärung der Frage, wie viele Teilmengen durchprobiert werden können, sodass der Aufwand für den Computer in einem aus unserer Sicht noch akzeptablen Rahmen bleibt, ist es an dieser Stelle nach der abstrakten Berechnungsmethode hilfreich, wenn konkrete Zahlen berechnet werden. Bis zur Mächtigkeit 3 könnten das die SuS selbst machen, für den Rest ist es ausreichend, wenn sie mit den Zahlen konfrontiert werden.

Anzahl Gewichte	***Anzahl aller Teilmengen bis einschliesslich der Mächtigkeit …***				
	3	*4*	*5*	*6*	*7*
20	1‘351	6‘196	21‘700	60‘460	137‘980
50	20‘876	251‘176	2‘369‘936	18‘260‘636	118‘145‘036
100	166‘751	4‘087‘976	79‘375‘496	1‘271‘427‘896	17‘278‘988‘696
1‘000	166‘667‘501	41‘583‘792‘251	$8.29188 \cdot 10^{12}$	$1.37647 \cdot 10^{15}$	$1.95657 \cdot 10^{17}$
10‘000	$1.66667 \cdot 10^{11}$	$4.16583 \cdot 10^{14}$	$8.32917 \cdot 10^{17}$	$1.38764 \cdot 10^{21}$	$1.98135 \cdot 10^{24}$
100‘000	$1{,}66667 \cdot 10^{14}$	$4{,}16658 \cdot 10^{18}$	$8{,}33292 \cdot 10^{22}$	$1{,}38876 \cdot 10^{27}$	$1{,}98385 \cdot 10^{31}$

Aufgabe 11: *Diskutiert, bis zu welcher Anzahl von Elementen der Computer alle Teilmengen in einem verbesserten Algorithmus für das SKP durchprobieren soll, sodass der Aufwand für den Computer noch akzeptabel ist! Gibt es für diese Aufgabe eine eindeutige Lösung?*

Im Gespräch über Aufgabe 11 soll klar werden, **dass der Rechenaufwand beim Vergrössern der Mächtigkeit der Teilmengen, die alle durchprobiert werden, exponentiell steigt**, und dass auch die Länge der Eingabe (Anzahl der Gewichte) den Rechenaufwand massgeblich beeinflusst.

Es sollte auch klar werden, dass es keine für alle Instanzen gültige Antwort auf die Frage, bis zu welcher Mächtigkeit alle Teilmengen durchprobiert werden sollen, gibt. Das könnte noch zu der Idee führen, dass wir in unserem verbesserten Algorithmus per Parameter bestimmen, bis zu welcher Länge der Teilmengen wir eine optimale Lösung berechnen wollen (abhängig davon, wie gross die Anzahl der Gewichte ist und wie viel Rechenaufwand für uns noch akzeptabel ist).

WIE GUT IST DER VERBESSERTE ALGORITHMUS FÜR DAS SKP?

Zu welchem Anteil wird der Rucksack mit dem verbesserten Algorithmus auf jeden Fall mindestens gefüllt, abhängig von der Mächtigkeit der Teilmengen, bis zu der die optimalen Teillösungen berechnet werden?

Diese Frage ist sicher die „Königsaufgabe" und nur für sehr wenige SuS beantwortbar. Daher sollen die SuS mithilfe von Intuition und logischem Denken dazu geführt werden, bestimmen zu können, zu welchem Anteil die Rucksäcke jeweils mindestens gefüllt werden. Dabei soll schrittweise vorgegangen werden und die Gedanken sollen mit Abbildungen veranschaulicht werden.

GEDANKENGANG FÜR DIE BESTIMMUNG DER GÜTE DES VERBESSERTEN ALGORITHMUS

Wir gehen bei unserer Argumentation von einem Lösungskandidaten aus, der auf jeden Fall betrachtet wird, wenn wir die beste Lösung aus einer Teilmenge mit höchstens k Objekten und der Greedy-Erweiterung berechnen.

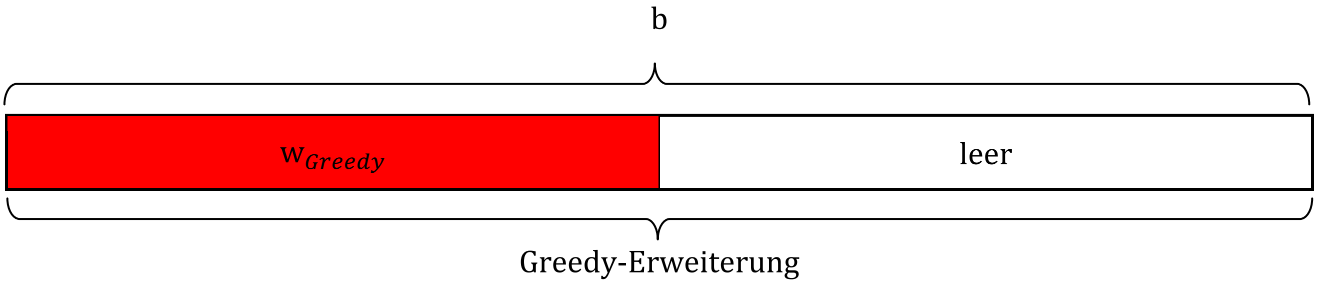
Wir haben:

- (mindestens) eine **optimale Lösung für die gegebene Instanz** mit z Gewichten: $\mathrm{w}_{Opt_1}, \mathrm{w}_{Opt_2}, \dots\ \mathrm{w}_{Opt_z}$
- alle **Lösungskandidaten**, die unser verbesserter Algorithmus betrachtet. Diese bestehen aus:
  - einer Teilmenge mit bis zu k Gewichten
  - einem oder mehreren Gewichten der Greedy-Erweiterung: $\mathrm{w}_{Greedy_1}, \mathrm{w}_{Greedy_2}, \dots$
  - einem „Rest" an freiem Platz im Rucksack, bezeichnet mit dem Wort „leer" (ausgenommen den Fall, dass die optimale Lösung den Rucksack zur Gänze füllt und der betrachtete Lösungskandidat der optimalen Lösung entspricht)

Wir wissen:

- Im schlechtesten Fall ist die mit der Teilmenge aus k Elementen und ihrer Greedy-Erweiterung erreichte beste Lösung **nicht gleich der optimalen Lösung** für die gegebene Instanz. Es gibt also mit Sicherheit noch einen Teil des Rucksacks, der leer bleibt.
- Wenn wir die Greedy-Erweiterung aller Lösungskandidaten als eigenständiges SKP betrachten, das mit der Greedy-Methode gelöst wird, wird damit mit Sicherheit **mindestens die Hälfte des freien Platzes**, der neben der Teilmenge mit k Gewichten noch übrigbleibt, **gefüllt**.
- Der schlechteste Fall ist, wenn in der Greedy-Erweiterung **nur noch genau ein Gewicht** eingepackt werden kann (siehe Erklärung zur denkbar schlechtesten Lösung mit der Greedy-Methode bei der Besprechung von Aufgabe 9).

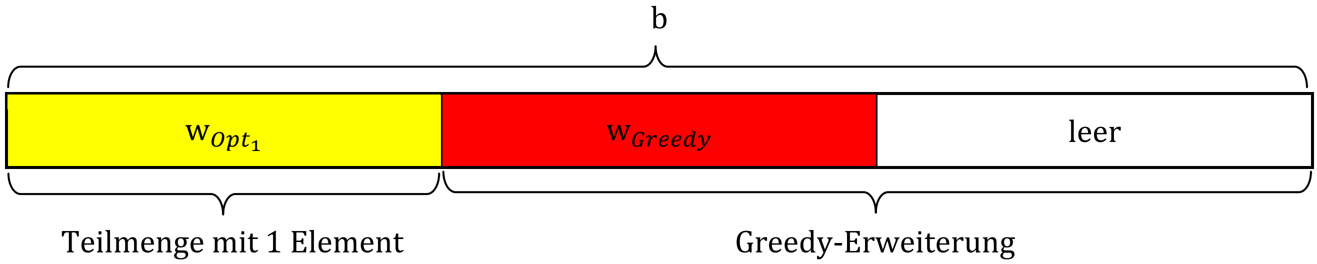
Beginnen wir nun mit dem (zugegebenermassen unsinnig erscheinenden) Fall der besten Lösung **mit einer Teilmenge mit 0 Elementen**. Dann wird der gesamte Rucksack mit der Greedy-Erweiterung gefüllt. Wie schon gezeigt wurde, erreichen wir damit die Approximationsgüte 2, oder in anderen Worten: Der Rucksack wird mindestens halbvoll.



Wie sieht es aus, wenn wir die beste Lösung mit einer Teilmenge **mit 1 Element** gefunden haben?

Falls die optimale Lösung für die gegebene Instanz nur aus einem Objekt besteht, entspricht die mit unserem Algorithmus erreichte beste Lösung der optimalen Lösung und wir sind fertig. Sehen wir uns daher jetzt den Fall an, dass die optimale Lösung aus mindestens zwei Objekten besteht.

Der Algorithmus betrachtet **alle** Teilmengen mit einem Element, erweitert sie mit der Greedy-Methode und nimmt als Lösung den besten von den n erhaltenen Lösungskandidaten.



Somit betrachtet der Algorithmus sicher auch das Objekt w_{Opt_1}, welches das grösste Gewicht aus einer optimalen Lösung ist. Wenn durch die Greedy-Erweiterung auf diesem Objekt die optimale Lösung erzeugt wird, sind wir fertig. Falls nicht, gibt es nach der Greedy-Erweiterung mindestens ein Gewicht w_{Opt_2} aus der optimalen Lösung, das nicht in diesem Lösungskandidaten des Algorithmus enthalten ist.

Von diesem Gewicht w_{Opt_2} wissen wir, dass es grösser als leer sein muss, weil der Algorithmus ansonsten w_{Opt_2} eingepackt hätte. (Das gilt auch für alle anderen noch nicht eingepackten Gewichte.)

Vom Objekt mit dem grössten Gewicht w_{Opt_1} aus der optimalen Lösung wissen wir, dass es grösser gleich w_{Opt_2} ist. Somit ist w_{Opt_1} auch grösser als leer.

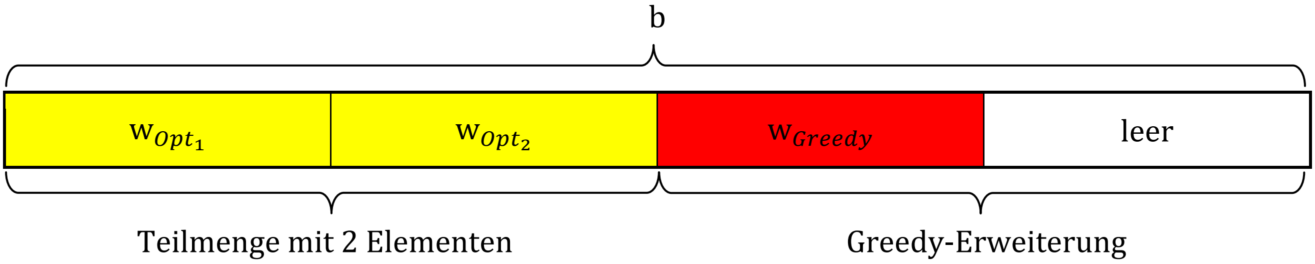
Vom Teil w_{Greedy}, der bei der Greedy-Erweiterung hinzugefügt wird, wissen wir, dass er auch grösser als leer ist, weil man mit der Greedy-Methode immer mindestens (minimal mehr als) die Hälfte des noch verbleibenden freien Platzes füllt und die Menge der Objekte, die man bei der Greedy-Erweiterung einpacken kann, nicht leer ist, da die optimale Lösung aus mindestens zwei Objekten besteht.

Somit können wir für diesen Lösungskandidaten sagen, dass leer kleiner als ein Drittel der Füllgrenze b ist. Der Rucksack ist mit Sicherheit zu $\frac{2}{3}$ gefüllt. Die Approximationsgüte des Algorithmus beträgt 1.5.

Wir gehen einen Schritt weiter und sehen uns die Situation an, bei der wir die beste Lösung mit einer Teilmenge **mit 2 Elementen** gefunden haben.

Auch hier gilt: Falls die optimale Lösung für die gegebene Instanz nur aus einem Objekt oder zwei Objekten besteht, haben wir die optimale Lösung erreicht. Sehen wir uns daher den Fall an, dass die optimale Lösung aus mindestens drei Objekten besteht.

Der Algorithmus betrachtet **alle** Teilmengen mit zwei Elementen, erweitert sie mit der Greedy-Methode und nimmt den besten von den $\frac{n \cdot (n-1)}{2}$ erhaltenen Lösungskandidaten.



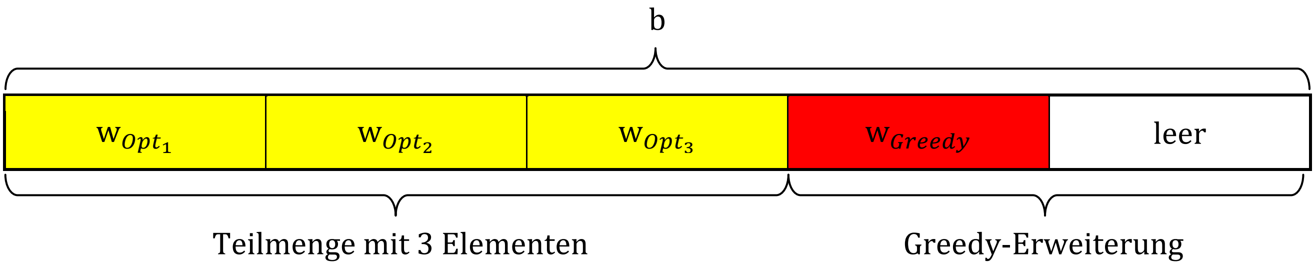
Somit betrachtet der Algorithmus sicherlich auch die Teilmenge mit den Gewichten w_{Opt_1} und w_{Opt_2}, den grössten Gewichten aus einer optimalen Lösung. Wenn durch die Greedy-Erweiterung dann die optimale Lösung nicht erzeugt werden kann, gibt es mindestens ein Gewicht w_{Opt_3} aus der optimalen Lösung, das nicht in diesem Lösungskandidaten des Algorithmus enthalten ist.

Dieses Gewicht w_{Opt_3} muss grösser als leer sein. Sowohl w_{Opt_1} als auch w_{Opt_2} sind grösser gleich w_{Opt_3} und daher ebenfalls grösser als leer. w_{Greedy_1} ist auch grösser als leer, weil mit diesem Objekt mindestens (minimal mehr als) die Hälfte des noch verbleibenden freien Platzes gefüllt wurde.

Somit ist leer bei diesem Lösungskandidaten kleiner als ein Viertel der Füllgrenze b und der Rucksack mit Sicherheit zu $\frac{3}{4}$ gefüllt. Die Approximationsgüte beträgt 1.33.

Zu guter Letzt sehen wir uns noch die Situation an, bei der wir die beste Lösung mit einer Teilmenge **mit 3 Elementen** gefunden haben. Wir betrachten den Fall, dass die optimale Lösung nicht aus drei, sondern mindestens aus vier Elementen besteht.

Der Algorithmus betrachtet **alle** Teilmengen mit drei Elementen, erweitert sie mit der Greedy-Methode und nimmt den besten von den $\frac{n \cdot (n-1) \cdot (n-2)}{6}$ erhaltenen Lösungskandidaten.



Es ist offensichtlich, dass das, was vorhin festgestellt wurde, analog auch in dieser Situation gilt. Unter allen Lösungskandidaten wird auch die Teilmenge mit den Gewichten w_{Opt_1}, w_{Opt_2} und w_{Opt_3}, den grössten Gewichten aus einer optimalen Lösung, betrachtet. w_{Opt_4} ist kleiner als diese drei Gewichte, aber grösser als leer. w_{Greedy_1} ist auch grösser als leer.

Somit ist leer bei diesem Lösungskandidaten kleiner als ein Fünftel der Füllgrenze b und der Rucksack mit Sicherheit zu $\frac{4}{5}$ gefüllt. Die Approximationsgüte beträgt 1.25.

Aufgabe 12: *Bis zu welchem Anteil wird der Rucksack mit dem verbesserten Algorithmus auf jeden Fall mindestens gefüllt, wenn die Anzahl der Elemente der Teilmengen, bis zu der die optimalen Teillösungen berechnet werden, mit 4 bestimmt wird?*

Als Lösung dieser Aufgabe kommt dem angeführten Gedankengang folgend heraus, dass der Rucksack mit Sicherheit zu mehr als $\frac{5}{6}$ gefüllt wird, die Approximationsgüte des Algorithmus also 1.2 ist.

Die letzte Erkenntnis, die festgehalten werden soll, ist, dass **auch bei verhältnismässig geringer Erhöhung des Rechenaufwands die Approximationsgüte deutlich verbessert werden kann**.

Falls wir die Anzahl der Elemente der Teilmengen, bis zu der eine optimale Lösung berechnet wird, selbst wählen, ist die **Approximationsgüte nicht mehr konstant**. Je grösser die Anzahl der Elemente in den Teilmengen ist, bis zu der alle Lösungskandidaten durchprobiert werden, umso mehr nähert sich die Approximationsgüte dem Wert 1 an. Dadurch steigt allerdings auch der Rechenaufwand abhängig von der Grösse der Menge an Gewichten exponentiell, wie folgende Übersichten zeigen.

Dargelegt wird zunächst die Anzahl der generierten Lösungskandidaten im Vergleich zur erreichten Approximationsgüte, veranschaulicht an den drei Beispielgrössen $\mathrm{n} = 100$, $\mathrm{n} = 1'000$, $\mathrm{n} = 10'000$:

Anzahl k Elemente	**Anzahl der Lösungskandidaten**			**Approximationsgüte**
	$\mathrm{n} = 100$	$\mathrm{n} = 1'000$	$\mathrm{n} = 10000$	
0	1	1	1	2
1	101	1‘001	10‘001	1.5
2	5‘051	500‘501	50‘005‘001	$1.\overline{33}$
3	166‘751	166‘667‘501	$1.66667 \cdot 10^{11}$	1.25
4	4‘087‘976	41‘583‘792‘251	$4.16583 \cdot 10^{14}$	1.2
5	79‘375‘496	$8.29188 \cdot 10^{12}$	$8.32917 \cdot 10^{17}$	$1.1\overline{6}$

Im Weiteren wird allgemein formuliert:

Anzahl k Elemente	**Anzahl der Lösungskandidaten**	**Approximationsgüte**
0	1	$\frac{2}{1} = 2$
1	$+\frac{n}{1}$	$\frac{3}{2} = 1.5$
2	$+\frac{n \cdot (n-1)}{2 \cdot 1}$	$\frac{4}{3} = 1.\overline{33}$
3	$+\frac{n \cdot (n-1) \cdot (n-2)}{3 \cdot 2 \cdot 1}$	$\frac{5}{4} = 1.25$
4	$+\frac{n \cdot (n-1) \cdot (n-2) \cdot (n-3)}{4 \cdot 3 \cdot 2 \cdot 1}$	$\frac{6}{5} = 1.2$

Aufgabe 13: *Formuliert jeweils eine allgemeingültige Formel für die Approximationsgüte als Funktion von* k *und für die Anzahl der Lösungskandidaten als Funktion von* k *abhängig von der Eingabegrösse* n*!*

Aus dem Gedankengang zur Bestimmung der Güte des verbesserten Algorithmus und den Werten in den oben angeführten Tabellen ergeben sich folgende Formeln:

- für die Approximationsgüte: $R_{A_{verb.}}(k) = \frac{k+2}{k+1}$
- für die Anzahl der Lösungskandidaten[4]:
  $T(k,n) = 1 + \frac{n}{1} + \frac{n \cdot (n-1)}{2 \cdot 1} + \cdots + \frac{n \cdot (n-1) \cdot \ldots \cdot (n-(k-2)) \cdot (n-(k-1))}{k \cdot (k-1) \cdot \ldots \cdot 2 \cdot 1}$

Es befinden sich im Zähler des letzten Summanden von $T(k,n)$ also k Zahlen im Produkt. Daher kann man, um den SuS eine einfache Vorstellung über die Anzahl der zu betrachtenden Lösungskandidaten zu offerieren, sagen, dass $T(k,n)$ ungefähr so schnell wächst wie n^k.

Diese letzte Aufgabe ist lediglich als Bonusaufgabe für die Leistungsstarken unter den SuS zu verstehen und ebenso optional wie die Geschichte zum Einstieg ganz zu Beginn der Unterrichtseinheit. Entscheidend ist als Abschluss, dass die SuS sehen, **mit wie viel Mehraufwand man die verbesserte Approximationsgüte bezahlt.**

[4] Die hier dargestellte Formel ist identisch mit $\sum_{i=0}^{k} \binom{n}{i}$. Allerdings gehen wir davon aus, dass die SuS weder das Summenzeichen noch das Berechnen von Kombinationen ohne Wiederholungen („n tief k“) kennen.