

Das Prinzip der Induktion als Problemlösestrategie und zur Entwicklung effizienter Algorithmen

Lernziele



Ziele

Das Prinzip der Induktion ist eine in der Informatik häufig erfolgreiche Problemlösestrategie. Dabei wird die Lösung eines Problems mit der Eingabegrösse n zurückgeführt auf jene des Problems mit der Eingabegrösse $n - 1$. Damit handelt es sich bei der Induktion um einen Spezialfall des allgemeineren Prinzips der Reduktion. Bei Letzterem wird die Lösung eines komplexen Problems auf jene eines Teilproblems oder mehrerer, einfacherer Teilprobleme zurückgeführt. Die Idee der Induktion bildet die Grundlage für wichtige Strategien in der Informatik wie die Rekursion und die dynamische Programmierung.

In dieser Lerneinheit soll das Prinzip der Induktion anhand zweier Beispiele vorgestellt werden. In einem ersten, geometrischen Beispiel wird die Idee des Zurückführens auf ein Teilproblem anschaulich eingeführt. In einem zweiten Beispiel erfahren die Schüler*innen, wie man mit der Idee der Induktion einen effizienten Algorithmus für die Lösung eines vereinfachten Rucksackproblems entwickeln kann.

Voraussetzungen



Voraussetzung

Die Einheit richtet sich an Lernende der 10. Klasse.

Aus dem Fachbereich Mathematik wird vorausgesetzt, dass die Schüler*innen elementare Termumformungen beherrschen, den Umgang mit Indizes und den «kleinen Gauss» kennen. Ausserdem wissen sie, was man unter konvexen Vielecken versteht.

Teil 1 – Die Idee der Induktion am Beispiel der Anzahl Diagonalen in einem konvexen n -Eck

Das Ziel der folgenden Aufgaben ist, eine Formel für die Anzahl Diagonalen in einem konvexen n -Eck herzuleiten. Dabei werden Sie eine Problemlösestrategie kennenlernen, die in der Informatik häufig erfolgreich eingesetzt wird.

Lernaufgaben

Andrea und Sam diskutieren darüber, wie viele Diagonalen ein konvexes Siebeneck hat. Andrea weiss, dass ein Sechseck neun Diagonalen hat. Da meint Sam: «Beim Siebeneck sind es natürlich fünf mehr, also insgesamt 14 Diagonalen!»

1) Erklären Sie, wie Sam darauf kommt, dass das Siebeneck fünf Diagonalen mehr als das Sechseck hat. Tipp: Skizzieren Sie ein Sechseck und überlegen Sie sich, welche Diagonalen hinzukommen, wenn man eine zusätzliche Ecke einfügt und aus dem Sechseck ein Siebeneck macht.

- 2) Das Prinzip aus Aufgabe 1) kann auch für andere Eckenzahlen angewendet werden. So hat ein Zehneck zum Beispiel 35 Diagonalen. Wie viele Diagonalen hat dann ein Elfeck?
- 3) Ein 101-Eck hat mehr Diagonalen als ein 100-Eck. Um wie viele Diagonalen erhöht sich die Anzahl der Diagonalen?
- 4) Sei nun $D(n)$ die Anzahl Diagonalen in einem konvexen n -Eck. Vervollständigen Sie die untenstehende Formel, welche das Resultat aus Aufgabe 3) darstellen soll:

$$D(101) = D(100) + \underline{\hspace{2cm}}$$

- 5) Geben Sie allgemein eine Formel an, mit der man die Anzahl Diagonalen eines $(n + 1)$ -Ecks aus jener eines n -Ecks berechnen kann:

$$D(n + 1) = D(n) + \underline{\hspace{2cm}}$$

- 6) Nutzen Sie die Formel aus Aufgabe 5) und füllen Sie die folgende Tabelle von links nach rechts aus:

Anzahl Ecken n	3	4	5	6	7	8	9	10	11
Anzahl Diagonalen $D(n)$	0								

- 7) Andrea und Sam stellen fest, dass man die Anzahl Diagonalen des Siebenecks auch folgendermassen hätte berechnen können: $D(7) = 2 + 3 + 4 + 5$. Welche Überlegung liegt hier zugrunde?
- 8) Mit welcher Summe analog zu Aufgabe 7) könnte man $D(n)$ berechnen?
- 9) Vereinfachen Sie die Summe aus Aufgabe 8), indem Sie den «kleinen Gauss» anwenden:

$$1 + 2 + 3 + \dots + k = \frac{k(k + 1)}{2}$$

Teil 2 – Das Wurstsack-Problem: Mit Induktion effiziente Algorithmen entwickeln

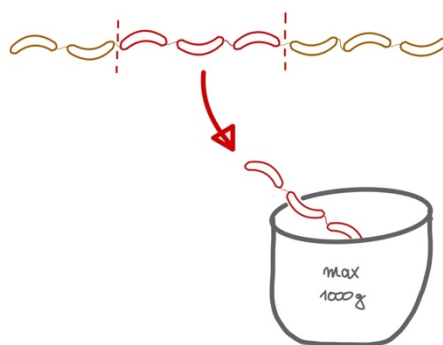
Einführungsaufgaben

In den folgenden Aufgaben wird das «Wurstsack-Problem» untersucht:

Würste mit unterschiedlichem Gewicht sind in einem Strang verbunden.



Ein Sack soll nun mit einem **zusammenhängenden Teilstrang** gefüllt werden, sodass die Summe der Gewichte der eingepackten Würste möglichst gross ist. Allerdings darf die Summe die Belastungsgrenze von 1000g nicht übersteigen, ansonsten reisst der Sack:



- 1) Betrachten Sie den folgenden Wurststrang. Die Zahlen geben das Gewicht der einzelnen Würste in Gramm an. Sie betragen von links nach rechts 250g, 500g, 50g, 500g, 50g, 100g, 150g und 300g.



Welcher zusammenhängende Teilstrang müsste eingepackt werden, damit das Totalgewicht möglichst gross, jedoch nicht über 1000g ist? Was wäre dann das Gesamtgewicht der eingepackten Würste?

- 2) Diskutieren Sie zu zweit: Wie sind Sie bei Aufgabe 1 vorgegangen?

- 3) Betrachten Sie nun einen neuen Wurststrang:



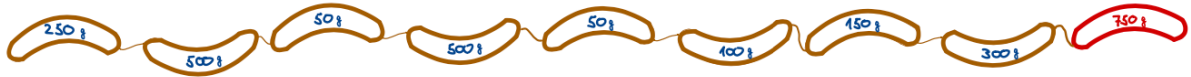
Die ersten acht Würste haben genau die gleichen Gewichte wie jene in der ersten Aufgabe. Neu wurde am Ende jedoch eine Wurst mit dem Gewicht 300g hinzugefügt. Welcher Teilstrang würde jetzt in den Sack eingepackt?

- 4) Welcher Teilstrang wird eingepackt, wenn die zusätzliche Wurst ein Gewicht von 480g hat?



¹ Bild aus einer Ansichtskarte, Urheber unbekannt

- 5) Welcher Teilstrang wird eingepackt, wenn die zusätzliche Wurst 750g wiegt?



- 6) Diskutieren Sie wieder zu zweit: Mit welcher Strategie haben Sie die Aufgaben 3), 4) und 5) gelöst?



Im Folgenden sollen unterschiedliche Lösungsstrategien für das Wurstsack-Problem verglichen werden. Zu diesem Zweck ist es praktischer, wenn die Fragestellung abstrahiert wird und hilfreiche Bezeichnungen eingeführt werden:

Das Wurstsack-Problem:

Gegeben: $n + 1$ positive Zahlen $w_1, w_2, \dots, w_n$ und B .

Gesucht: Zwei Zahlen $i \leq j \in \{1, 2, \dots, n\}$, sodass $S_{ij} = w_i + w_{i+1} + \dots + w_j \leq B$ und möglichst gross ist.

Die Zahlen w_1 bis w_n stehen für die Gewichte der Würste und B steht für die Belastungsgrenze des Sacks. Die gesuchten Zahlen i und j bezeichnen die Position der Anfangs- und Endwurst des optimalen Teilstrangs. Das Gewicht des Wurststrangs, der mit Wurst Nummer i beginnt und mit Nummer j endet, wird mit S_{ij} bezeichnet.

Auch Sam und Andrea haben sich mit dem Wurstsack-Problem befasst und unterschiedliche Lösungsstrategien entwickelt:

Sams Strategie: Brute-Force

Sam berechnet alle Summen S_{ij} mit $1 \leq i \leq j \leq n$. Am Schluss werden die Summen verglichen und es wird jene ermittelt, welche möglichst gross ist, aber B nicht überschreitet: Am Beispiel des Wurststrangs von Aufgabe 1 sähe Sams Strategie so aus:



- $i = 1$:
 - $S_{11} = 250$
 - $S_{12} = 250 + 500 = 750$
 - $S_{13} = 250 + 500 + 50 = 800$
 - $S_{14} = 250 + 500 + 50 + 500 = 1300$
 - $S_{15} = 250 + 500 + 50 + 500 + 50 = 1350$
 - $S_{16} = 250 + 500 + 50 + 500 + 50 + 100 = 1450$
 - $S_{17} = 250 + 500 + 50 + 500 + 50 + 100 + 150 = 1600$
 - $S_{18} = 250 + 500 + 50 + 500 + 50 + 100 + 150 + 300 = 1900$
- $i = 2$
 - $S_{22} = 500$
 - $S_{23} = 500 + 50 = 550$
 - $S_{24} = 500 + 50 + 500 = 1050$
 - $S_{25} = 500 + 50 + 500 + 50 = 1100$
 - $S_{26} = 500 + 50 + 500 + 50 + 100 = 1200$
 - $S_{27} = 500 + 50 + 500 + 50 + 100 + 150 = 1350$
 - $S_{28} = 500 + 50 + 500 + 50 + 100 + 150 + 300 = 1650$
- ... Analog werden die Startwürste $i = 3$ bis $i = 6$ untersucht ...
- $i = 7$
 - $S_{77} = 150$
 - $S_{78} = 150 + 300 = 450$

- $i = 8$
 - $S_{88} = 300$

Schliesslich wird die grösste aller Gewichtssummen, die jedoch kleiner als die Belastungsgrenze $M = 1000$ ist, ermittelt. So erhält Sam die gesuchte Lösung $i = 3, j = 7$ mit $S_{37} = 850$.

Aufgaben

In diesen Aufgaben soll untersucht werden, wie effizient Sams Lösungsstrategie ist. Dazu überlegen wir, wie viele Additionen nötig sind, um das Resultat zu finden.

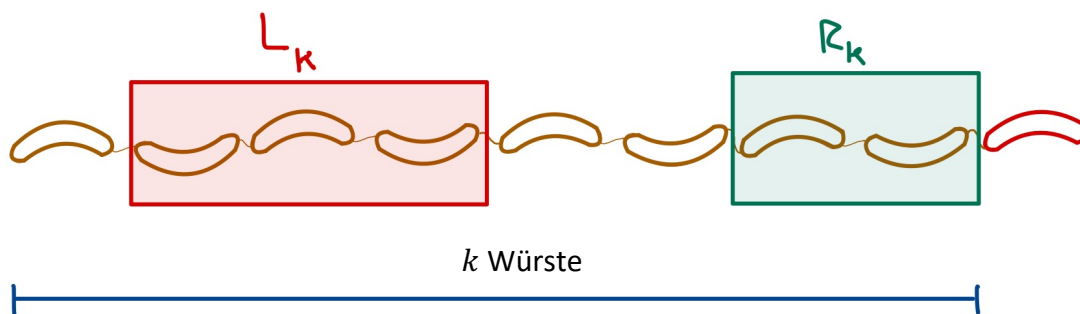
- 1) Betrachten Sie nochmals Sams Strategie am Beispiel des Strangs aus acht Würsten oben.
 - a) Wie viele Additionen werden benötigt, um die «Startwurst $i = 1$ » zu untersuchen?
Tipp: Zählen Sie die Anzahl Pluszeichen.
 - b) Wie viele Additionen sind es für alle Stränge mit der Startwurst $i = 2$, wie viele für Startwurst $i = 3$, wie viele Additionen sind es insgesamt?
- 2) Angenommen der Strang besteht aus n Würsten:
 - a) Wie viele Additionen benötigt man für Startwurst Nr. 1, Nr. 2, Nr. 3, usw.?
 - b) Challenge: Nutzen Sie den «kleinen Gauss» und die folgende Formel, um die gesamte Anzahl erforderlicher Additionen zu vereinfachen:

$$1 \cdot 2 + 2 \cdot 3 + 3 \cdot 4 + \dots + (k - 1) \cdot k = \frac{1}{3} \cdot (k - 1)k(k + 1)$$
- 3) Wie könnte man Sams Strategie verbessern, sodass weniger Additionen nötig sind?

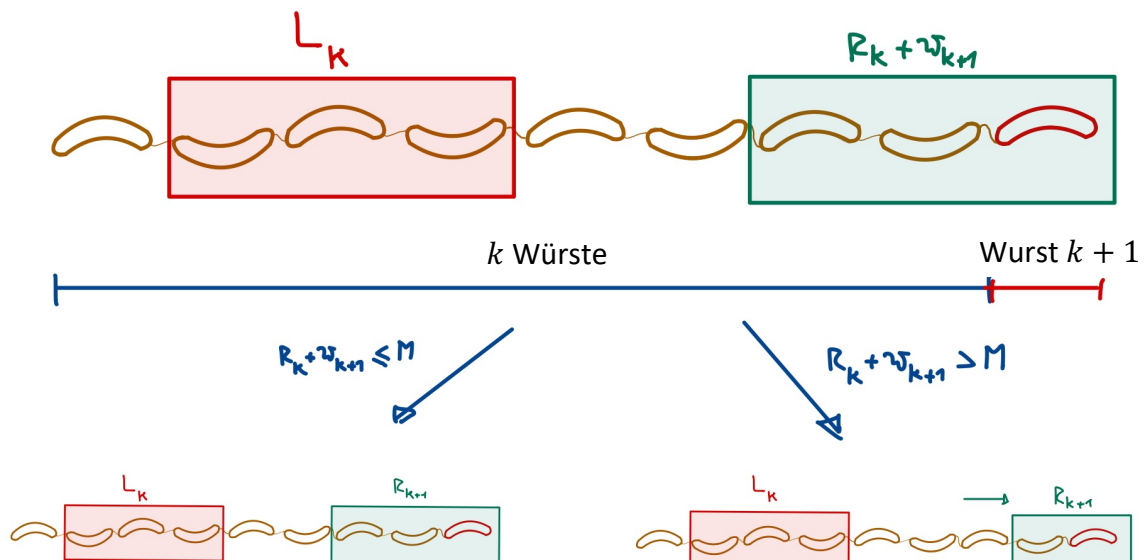


Andrea's Strategie: Induktion

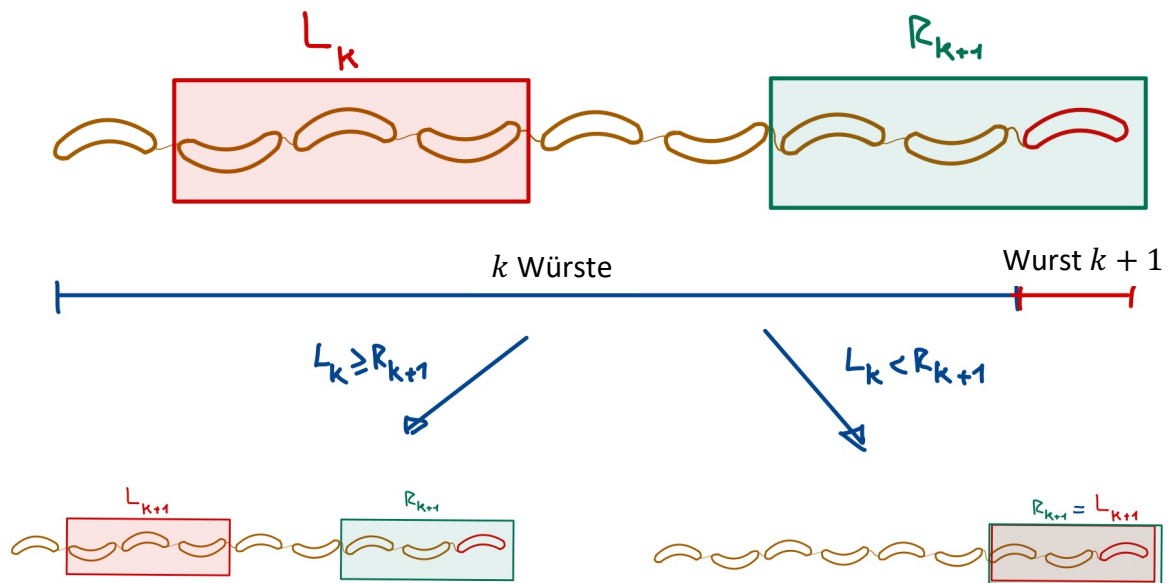
Andrea hat sich bei der Entwicklung einer Lösungsstrategie vom Problem der Diagonalen im Rechteck inspirieren lassen. Die Grundidee ist dabei wie folgt: Wenn man die optimale Lösung des Problems für die ersten k Würste des Strangs kennt, kann man daraus leicht die Lösung für die ersten $(k + 1)$ Würste ermitteln. Andrea löst also der Reihe nach das Wurstsack-Problem für die erste Wurst, für die ersten beiden Würste, für die ersten drei Würste und so weiter. Dabei notiert Andrea sich für jedes Teilproblem die gefundene optimale Lösung L_k und die grösste zulässige Gewichtssumme R_k eines Strangs «rechts», welcher die zuletzt hinzugefügte Wurst beinhaltet.



Beim Übergang auf das nächstgrössere Teilproblem – bestehend aus den ersten $(k + 1)$ Würsten – ermittelt Andrea die grösste zulässige Summe R_{k+1} , welche Wurst Nr. $(k + 1)$ beinhaltet. Dazu erweitert man den Teilstrang R_k um die $(k + 1)$ -ste Wurst und prüft, ob damit die Belastungsgrenze überschritten wird: $R_k + w_{k+1} > M$? Falls dies nicht der Fall ist, setzt man $R_{k+1} = R_k + w_{k+1}$. Ansonsten schneidet Andrea auf der «linken» Seite von $R_k + w_{k+1}$ so lange Würste ab, bis das Gewicht wieder unter M fällt.



Der grössere der beiden Werte L_k und R_{k+1} bildet dann die Lösung L_{k+1} .



Am Beispiel-Wurststrang sähe Andrea's Strategie so aus:



- Teilproblem bestehend aus der ersten Wurst:
 - $w_1 = 250 < 1000$
 - $\rightarrow$ Lösung des Teilproblems $L_1 = S_{11} = 250$
 - $\rightarrow$ Grösste zulässige Summe, welche Wurst Nr. 1 beinhaltet $R_1 = S_{11} = 250$
- Teilproblem bestehend aus den ersten zwei Würsten:
 - $R_1 + w_2 = 250 + 500 = 750 < 1000 \Rightarrow R_2 = S_{12} = 750$
 - $L_1 < R_2 \Rightarrow L_2 = R_2 = S_{12} = 750$
- Teilproblem bestehend aus den ersten drei Würsten:
 - $R_2 + w_3 = 750 + 50 = 800 < 1000 \Rightarrow R_3 = S_{13} = 800$
 - $L_2 < R_3 \Rightarrow L_3 = R_3 = S_{13} = 800$
- Teilproblem bestehend aus den ersten vier Würsten:
 - $R_3 + w_4 = 800 + 500 = 1300 > 1000 \rightarrow$ Würste am «linken Ende» abschneiden.
 - $R_3 + w_4 - w_1 = 1300 - 250 = 1050 > 1000$
 - $R_3 + w_4 - w_1 - w_2 = 1050 - 500 = 550 < 1000$
 $\Rightarrow R_4 = S_{34} = 550$
 - $L_3 > R_4 \Rightarrow L_4 = L_3 = S_{13} = 800$
- Teilproblem bestehend aus den ersten fünf Würsten:
 - $R_4 + w_5 = 550 + 50 = 600 < 1000 \Rightarrow R_5 = S_{35} = 600$
 - $L_4 > R_5 \Rightarrow L_5 = L_4 = S_{13} = 800$
- Teilproblem bestehend aus den ersten sechs Würsten:
 - $R_5 + w_6 = 600 + 100 = 700 < 1000 \Rightarrow R_6 = S_{36} = 700$
 - $L_5 > R_6 \Rightarrow L_6 = L_5 = S_{13} = 800$
- Teilproblem bestehend aus den ersten sieben Würsten:
 - $R_6 + w_7 = 700 + 150 = 850 < 1000 \Rightarrow R_7 = S_{37} = 850$
 - $L_6 < R_7 \Rightarrow L_7 = R_7 = S_{37} = 850$
- Teilproblem bestehend aus den ersten acht Würsten:

- $R_7 + w_8 = 850 + 300 = 1150 > 1000 \rightarrow$ Würste am «linken Ende» abschneiden.
 - $R_7 + w_8 - w_3 = 1150 - 50 = 1100 > 1000$
 - $R_7 + w_8 - w_3 - w_4 = 1100 - 500 = 600 < 1000$
 $\Rightarrow R_8 = S_{58} = 600$
- $L_7 > R_8 \Rightarrow L_8 = L_7 = S_{37} = 850$

Fazit: Der schwerste zusammenhängende Teilstrang, welcher die Belastungsgrenze von 1000g nicht überschreitet, hat ein Gesamtgewicht von 850g und setzt sich aus den Würsten 3 bis 7 zusammen.

Aufgabe

- 1) Machen Sie sich mit Andrea's Strategie vertraut, indem Sie sie auf den folgenden Wurststrang anwenden:



Die Gewichte der Würste sind der Reihe nach: 100g, 600g, 450g, 200g, 150g, 50g, 400g, 500g.

Schon anhand der beiden ausgeführten Beispiele ahnt man, dass Andrea's Methode weniger aufwendig als jene von Sam ist. Im Hinblick auf einen genaueren Vergleich der beiden Strategien überlegen wir wieder, wie viele Additionen und Subtraktionen höchstens nötig sind, um mit Andrea's Strategie das Wurstsack-Problem mit n Würstchen zu lösen.

Die einzige Stelle, an welcher Additionen und Subtraktionen durchgeführt werden, ist die bei der Berechnung von R_{k+1} aus R_k . In jedem Fall wird dabei zuerst $R_k + w_{k+1}$ gerechnet, wobei dies insgesamt $(n - 1)$ -mal geschieht. Tritt der Fall ein, dass $R_k + w_{k+1} > M$ ist, müssen ausserdem die Würste am «linken Ende» von R_k abgeschnitten werden, was einer Subtraktion entspricht. Dabei kann jede Wurst aber höchstens einmal abgeschnitten werden, womit insgesamt höchstens n Subtraktionen entstehen. Total werden mit Andrea's Methode also insgesamt höchstens $(n - 1) + n = 2n - 1$ Operationen benötigt.



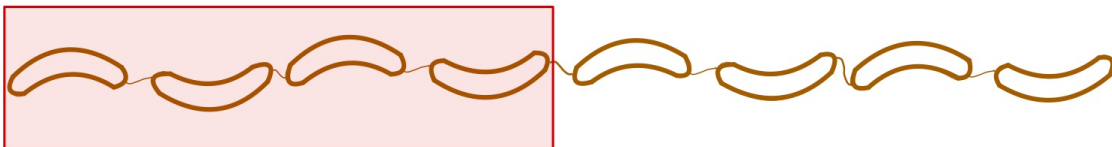
Eine dritte Strategie:

Beim Vergleich ihrer beiden Strategien entwickeln Sam und Andrea die Idee für eine dritte Strategie. Dabei werden wie bei Sam die Summen der Form S_{ij} berechnet – diesmal aber nur solche, die als Lösung des Problems infrage kommen. Ausserdem wird Andrea's Strategie zur Berechnung von R_{k+1} angepasst und für die Berechnung der S_{ij} übernommen.

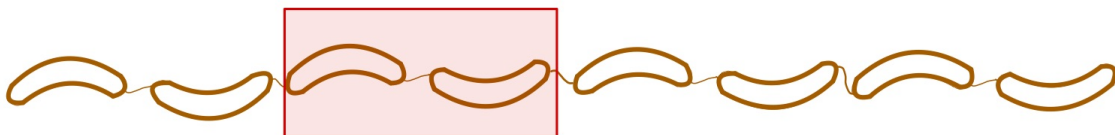
Genauer funktioniert dann die Strategie wie folgt: Man beginnt mit dem Teilstrang, der nur aus der ersten Wurst besteht: $i = j = 1$.



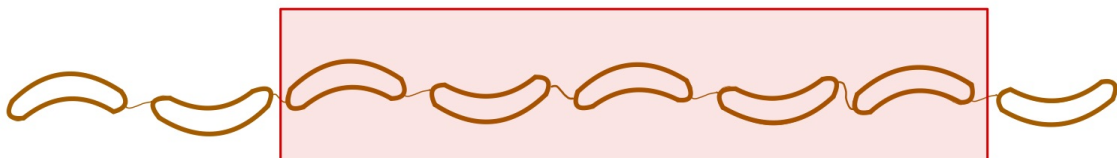
Anschliessend wird j so lange erhöht, bis S_{ij} erstmals die Belastungsgrenze B sprengt. Dabei nutzt man aus, dass jede Summe mit wenig Aufwand aus der vorhergehenden berechnet werden kann, indem man lediglich das Gewicht der neu dazugekommenen Wurst addiert.



Sobald die Belastungsgrenze überschritten wurde, wird i erhöht, bis die Summe S_{ij} wieder unter B fällt. Auch hier nutzt man aus, dass man die Summen durch geschicktes Vorgehen mit wenig Aufwand berechnen kann.



Wenn S_{ij} wieder unter B liegt, erhöht man wieder j .



Solange $j \leq n$ ist, fährt man fort, indem abwechselungsweise mal i und mal j erhöht werden. Am Beispiel-Wurststrang würde man konkret wie folgt vorgehen:



- $i = 1, j = 1, j$ schrittweise erhöhen:
 - $S_{11} = 250 < 1000$
 - $S_{12} = S_{11} + w_2 = 250 + 500 = 750 < 1000$
 - $S_{13} = S_{12} + w_3 = 750 + 50 = 800 < 1000$
 - $S_{14} = S_{13} + w_4 = 800 + 500 = 1300 > 1000$

Sobald die Belastungsgrenze überschritten wurde, wird i erhöht, bis die Summe S_{ij} wieder unter B fällt. Auch hier nutzt man aus, dass man die Summen durch geschicktes Vorgehen mit wenig Aufwand berechnen kann.

- $i = 1, j = 4, i$ erhöhen:
 - $S_{24} = S_{14} - w_1 = 1300 - 250 = 1050 > 1000$
 - $S_{34} = S_{24} - w_2 = 1050 - 500 = 550 < 1000$

Nun wird j wieder schrittweise erhöht:

- $i = 3, j = 4, j$ erhöhen:
 - $S_{35} = S_{34} + w_5 = 600 < 1000$
 - $S_{36} = S_{35} + w_6 = 700 < 1000$
 - $S_{37} = S_{36} + w_7 = 850 < 1000$
 - $S_{38} = S_{37} + w_8 = 1150 > 1000$

Nach Überschreiten der Grenze B wird wieder i erhöht:

- $i = 3, j = 8, i$ erhöhen:
 - $S_{48} = S_{38} - w_3 = 1100 > 1000$
 - $S_{58} = S_{48} - w_4 = 600 < 1000$

Nun müsste man j wieder erhöhen, da aber mit $j = 8$ das Ende des Wurststrangs erreicht wurde, endet der Algorithmus. Die höchste zulässige Summe ist $S_{37} = 850$.

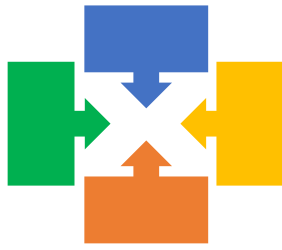
Aufgaben

1) Wenden Sie diese Methode für Würste mit den Gewichten 250g, 300g, 500g, 800g, 300g, 100g, 500g, 200g, 100g und der Belastungsgrenze $B = 1000g$ an.

Wieder wollen wir untersuchen, wie viele Additionen und Subtraktionen erforderlich sind, wenn man das Wurstsack-Problem mit dieser Methode löst.

- 2) Betrachten Sie nochmals die Strategie am Beispiel des Strangs aus acht Würsten.
 - a) Wie viele Additionen und Subtraktionen waren insgesamt erforderlich?
 - b) Bei wie vielen Operationen war das Gewicht w_2 von Würstchen Nummer 2 beteiligt und in wie vielen kam w_3 vor und in wie vielen w_4 und w_5 ?
- 3) Begründen Sie: Mit dieser Methode werden zur Lösung des Wurstsack-Problems mit n Würstchen höchstens $2n$ Operationen benötigt.

Zusammenfassung: Das Prinzip der Induktion

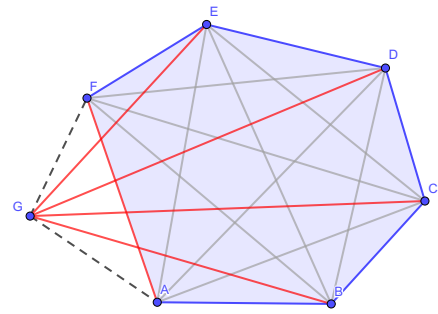


Im ersten Teil wurde eine Formel für die Anzahl Diagonalen eines konvexen Vielecks hergeleitet: Wie so oft war aber der Weg das eigentliche Ziel. Das Problem wurde mittels Induktion gelöst: Die Frage nach der Anzahl Diagonalen in einem n -Eck wurde auf jene der Anzahl Diagonalen im $(n - 1)$ -Eck zurückgeführt. Bei der Induktion handelt es sich um eine Problemlösungs-Strategie, die in der Informatik häufig zur Anwendung kommt. Dies wurde im zweiten Teil anhand des «Wurstsack-Problems» veranschaulicht. Andrea's Strategie verwendet das Prinzip der Induktion: Die Lösung für das Problem mit einem Strang aus n Würsten wird auf jenes des Teilstrangs aus den ersten $(n - 1)$ Würsten reduziert. Eine etwas ausgeklügelte Variante dieser Strategie – die dynamische Programmierung – wird auch zur Lösung des berühmten Rucksack-Problems angewendet.

Lösungen

Lösungen der Aufgaben von Teil 1:

1) Beachten Sie die Skizze rechts. Durch Hinzufügen einer weiteren Ecke G zum Sechseck $ABCDEF$ entsteht ein Siebeneck $ABCDEFG$. Die Diagonalen des Sechsecks (im Bild grau dargestellt) sind auch Diagonalen im Siebeneck. Zusätzlich besitzt das Siebeneck weitere fünf Diagonalen (im Bild mit roter Farbe gekennzeichnet) – eine für jede nicht mit G benachbarte Ecke und zusätzlich die Seite AF des Sechsecks, welche im Siebeneck eine Diagonale ist.



- 2) Wird ein Zehneck um eine Ecke erweitert, sodass ein Elfeck entsteht, kommen zu den bereits vorhandenen 35 Diagonalen einige hinzu:
- Eine für jede nicht mit der elften Ecke benachbarten Ecke des Zehnecks: also $10 - 2 = 8$ neue Diagonalen.
 - Zusätzlich wird eine Seite des Zehnecks im Elfeck zur Diagonale.
- Insgesamt hat ein Elfeck somit 9 Diagonalen mehr als ein Zehneck, also $35 + 9 = 44$ Diagonalen.
- 3) Analog zu der vorhergehenden Aufgabe hat das 101-Eck $(100 - 2) + 1 = 99$ Diagonalen mehr als das 100-Eck.

4) $D(101) = D(100) + 99$

5) $D(n + 1) = D(n) + \underbrace{(n - 2)}_{\substack{\text{Eine für jede} \\ \text{nicht benachbarte Ecke} \\ \text{des } n\text{-Ecks}}} + \underbrace{1}_{\substack{\text{eine Seite wird} \\ \text{zur Diagonale}}} = D(n) + n - 1$

6)

Anzahl Ecken n	3	4	5	6	7	8	9	10	11
Anzahl Diagonalen $D(n)$	0	2	5	9	14	20	27	35	44



- 7) Beachten Sie die Pfeile in der Lösung der Aufgabe 6). Wenn von links nach rechts gerechnet wird, folgt:

$$D(7) = 0 + 2 + 3 + 4 + 5$$

8) $D(n) = 2 + 3 + 4 + 5 + \dots + (n - 3) + (n - 2)$

9) $D(n) = 2 + 3 + 4 + 5 + \dots + (n - 3) + (n - 2)$
 $= 1 + 2 + 3 + 4 + 5 + \dots + (n - 3) + (n - 2) - 1$

Mit dem «kleinen Gauss» für $k = n - 2$ kann der Term oben vereinfacht werden zu:

$$D(n) = \frac{(n - 2)(n - 2 + 1)}{2} - 1 = \frac{(n - 2)(n - 1) - 2}{2} = \frac{n^2 - 3n}{2} = \frac{n(n - 3)}{2}$$



Lösungen der Aufgaben von Teil 2

Einführungsaufgaben

- 1) Es müssten die Würste 3 bis einschliesslich 7 mit einem Gesamtgewicht von 850g eingepackt werden.
- 2) Eine individuelle Vorgehensweise ist möglich.
- 3) Die optimale Lösung ist der Strang, der aus Wurst 5 bis einschliesslich Wurst 9 besteht.
- 4) Wieder gehört die neue Wurst zur optimalen Lösung, die diesmal aus den Würsten 7 bis einschliesslich 9 besteht.
- 5) Im Vergleich zur Situation aus Aufgabe 1 hat sich die optimale Lösung nicht verändert und besteht wieder aus den Würsten 3 bis einschliesslich 7.
- 6) Eine individuelle Vorgehensweise ist möglich, allerdings kann man ausnutzen, dass die Lösung des erweiterten Problems die neu dazugekommene Wurst beinhaltet oder der Lösung des ursprünglichen Problems entspricht.

Sams Methode

- 1) Es folgt die Effizienzanalyse von Sams Strategie am Beispiel des Strangs aus acht Würsten.

a) Für Startwurst $i = 1$ werden $1 + 2 + 3 + \dots + 7 = \frac{7 \cdot 8}{2} = 28$ Additionen benötigt.

b) Für $i = 2$ sind es $1 + 2 + \dots + 6 = \frac{6 \cdot 7}{2} = 21$ Additionen.

Für $i = 3$ $\frac{5 \cdot 6}{2} = 15$.

Insgesamt: $\frac{7 \cdot 8}{2} + \frac{6 \cdot 7}{2} + \frac{5 \cdot 6}{2} + \frac{4 \cdot 5}{2} + \frac{3 \cdot 4}{2} + \frac{2 \cdot 3}{2} + \frac{1 \cdot 2}{2} = 84$.

- 2) Angenommen der Strang besteht aus n Würsten:

a) Für $i = 1$: $\frac{(n-1)n}{2}$, für $i = 2$: $\frac{(n-2)(n-1)}{2}$ für $i = 3$: $\frac{(n-3)(n-2)}{2}$.

b) Challenge:

$$\begin{aligned} \frac{1 \cdot 2}{2} + \frac{2 \cdot 3}{2} + \dots + \frac{(n-1)n}{2} &= \frac{1}{2} (1 \cdot 2 + 2 \cdot 3 + 3 \cdot 4 + \dots + (n-1) \cdot n) \\ &= \frac{1}{6} (n-1)n(n-2) \end{aligned}$$

- 3) Für die Berechnung der einzelnen Summen reicht eine Addition, wenn man die vorhergehende Summe verwendet. So ist z.B. $S_{13} = S_{12} + w_3$.

Wenn die Belastungsgrenze einmal überschritten wurde, muss man die weiteren Teilstränge mit der gleichen Startwurst nicht mehr beachten. So muss man im aufgeführten Beispiel die Summen S_{15} bis S_{18} nicht berechnen.

Andrea's Methode:

- Teilproblem bestehend aus der ersten Wurst:
 - $w_1 = 100 < 1000$
 - $\rightarrow$ Lösung des Teilproblems $L_1 = S_{11} = 100$
 - $\rightarrow$ Grösste zulässige Summe, welche Wurst Nr. 1 beinhaltet $R_1 = S_{11} = 100$
- Teilproblem bestehend aus den ersten zwei Würsten:
 - $R_1 + w_2 = 100 + 600 = 700 < 1000 \Rightarrow R_2 = S_{12} = 700$
 - $L_1 < R_2 \Rightarrow L_2 = R_2 = S_{12} = 700$
- Teilproblem bestehend aus den ersten drei Würsten:
 - $R_2 + w_3 = 700 + 450 = 1150 > 1000 \rightarrow$ Würste am «linken Ende» abschneiden.
 - $R_2 + w_3 - w_1 = 1150 - 100 = 1050 > 1000$
 - $R_2 + w_3 - w_1 - w_2 = 1050 - 600 = 450 < 1000$
 $\Rightarrow R_3 = S_{33} = 450$
 - $L_2 > R_3 \Rightarrow L_3 = L_2 = S_{12} = 700$
- Teilproblem bestehend aus den ersten vier Würsten:
 - $R_3 + w_4 = 450 + 200 = 650 < 1000 \Rightarrow R_4 = S_{34} = 650$
 - $L_3 > R_4 \Rightarrow L_4 = L_3 = S_{12} = 700$
- Teilproblem bestehend aus den ersten fünf Würsten:
 - $R_4 + w_5 = 650 + 150 = 800 < 1000 \Rightarrow R_5 = S_{35} = 800$
 - $L_4 < R_5 \Rightarrow L_5 = R_5 = S_{35} = 800$
- Teilproblem bestehend aus den ersten sechs Würsten:
 - $R_5 + w_6 = 800 + 50 = 850 < 1000 \Rightarrow R_6 = S_{36} = 850$
 - $L_5 < R_6 \Rightarrow L_6 = R_6 = S_{36} = 850$
- Teilproblem bestehend aus den ersten sieben Würsten:
 - $R_6 + w_7 = 850 + 400 = 1250 > 1000 \rightarrow$ Würste am «linken Ende» abschneiden.
 - $R_6 + w_7 - w_3 = 1250 - 450 = 800 < 1000$
 $\Rightarrow R_7 = S_{47} = 800$
 - $L_6 > R_7 \Rightarrow L_7 = L_6 = S_{36} = 850$
- Teilproblem bestehend aus den ersten acht Würsten:
 - $R_7 + w_8 = 800 + 500 = 1300 > 1000 \rightarrow$ Würste am «linken Ende» abschneiden.
 - $R_7 + w_8 - w_4 = 1300 - 200 = 1100 > 1000$
 - $R_7 + w_8 - w_4 - w_5 = 1100 - 150 = 950 < 1000$
 $\Rightarrow R_8 = S_{68} = 950$
 - $L_7 < R_8 \Rightarrow L_8 = R_8 = S_{68} = 950$

Fazit: Der schwerste zusammenhängende Teilstrang, welcher die Belastungsgrenze von 1000g nicht überschreitet, hat ein Gesamtgewicht von 950g und setzt sich aus den Würsten 6 bis einschliesslich 8 zusammen.

Dritte Methode

1)

$$S_{11} = 250, S_{12} = 550, S_{13} = 1050;$$

$$S_{23} = 800, S_{24} = 1600;$$

$$S_{34} = 1300;$$

$$S_{44} = 800, S_{45} = 1100$$

$$S_{55} = 300, S_{56} = 400, S_{57} = 900, S_{58} = 1100;$$

$$S_{68} = 900$$

2) Wir betrachten Andrea's Strategie am Beispiel des Strangs aus acht Würsten.

a) Insgesamt sind 11 Operationen erforderlich.

b) Die Gewichte w_2 bis w_4 kamen je in einer Addition und in einer Subtraktion vor, die Gewichte w_5 bis w_8 nur je in einer Addition, das Gewicht w_1 kommt in einer Subtraktion vor.3) Mit jeder Wurst wird höchstens zweimal gerechnet. Einmal wird es beim Erweitern des Wurststrangs (Erhöhung von j) addiert und einmal beim Verkürzen des Strangs (Erhöhung von i) subtrahiert. Insgesamt gibt es also nicht mehr als $2n$ Operationen.