

Aufgabe 6 (Backtracking)

Ein sehr bekanntes Optimierungsproblem ist das sogenannte Rucksackproblem: Gegeben ist ein Rucksack mit der Tragfähigkeit B . Weiterhin ist eine endliche Menge von Gegenständen mit Werten und Gewichten gegeben. Nun soll eine Teilmenge der Gegenstände so ausgewählt werden, dass ihr Gesamtwert maximal ist, aber ihr Gesamtgewicht die Tragfähigkeit des Rucksacks nicht überschreitet.

Mathematisch exakt kann das Rucksackproblem wie folgt formuliert werden:

Gegeben ist eine endliche Menge von Objekten U . Durch eine Gewichtsfunktion $w: U \rightarrow \mathbb{R}^+$ wird den Objekten ein Gewicht und durch eine Nutzenfunktion $v: U \rightarrow \mathbb{R}^+$ ein festgelegter Nutzwert zugeordnet.

Des Weiteren gibt es eine vorgegebene Gewichtsschranke $B \in \mathbb{R}^+$. Gesucht ist eine Teilmenge $K \subseteq U$, die die Bedingung $\sum_{u \in K} w(u) \leq B$ einhält und die Zielfunktion $\sum_{u \in K} v(u)$ maximiert.

Das Rucksackproblem ist NP-vollständig (Problemgröße ist die Anzahl der Objekte), sodass es an dieser Stelle wenig Sinn macht, über eine effiziente Lösung nachzudenken. Lösen Sie das Rucksackproblem daher mittels Backtracking und formulieren Sie einen entsprechenden Algorithmus. Gehen Sie davon aus, dass die Gewichtsschranke B sowie die Anzahl an Objekten N beliebig, aber fest vorgegeben sind.

Das Programm soll folgende Ausgaben liefern:

- (a) Maximaler Nutzwert, der durch eine Objektauswahl unter Einhaltung der Gewichtsschranke B erreicht werden kann.
- (b) Das durch die maximierende Objektmenge erreichte Gesamtgewicht.
- (c) Diejenigen Objekte (Objektnummern) aus U , die zur Maximierung des Nutzwerts beigetragen haben.

```
3  /**
4   * https://stackoverflow.com/a/14186622
5   */
6  public class Rucksack {
7      // static int[] werte = new int[] { 894, 260, 392, 281, 27 };
8      // static int[] gewichte = new int[] { 8, 6, 4, 0, 21 };
9      // static int[] werte = new int[] { 4, 2, 10, 1, 2 };
10     // static int[] gewichte = new int[] { 12, 1, 4, 1, 2 };
11     static int werte[] = new int[] { 60, 100, 120 };
12     static int gewichte[] = new int[] { 10, 20, 30 };
13
14
15
16     /**
17      * Gewichtsschranke
18      */
19     //static int B = 30;
20     //static int B = 15;
21     static int B = 50;
22
23     /**
```

```

24      * Diejenigen Objekte aus U, die zur Maximierung des Nutzwerts beigetragen
25      * haben.
26      */
27      static boolean[] auswahl = new boolean[werte.length];
28
29      private static int berechne(int i, int W) {
30          if (i < 0) {
31              return 0;
32          }
33          int alt = berechne(i - 1, W);
34          if (gewichte[i] > W) {
35              // Backtracking!
36              auswahl[i] = false;
37              return alt;
38          } else {
39
40              int neu = berechne(i - 1, W - gewichte[i]) + werte[i];
41              if (alt >= neu) {
42                  // Backtracking!
43                  auswahl[i] = false;
44                  return alt;
45              } else {
46                  auswahl[i] = true;
47                  return neu;
48              }
49          }
50      }
51
52      static void werteAus() {
53          System.out.println(berechne(werte.length - 1, B));
54
55          int gesamtGewicht = 0;
56          int gesamtWert = 0;
57
58          for (int i = 0; i < auswahl.length; i++) {
59              if (auswahl[i]) {
60                  gesamtGewicht += gewichte[i];
61                  gesamtWert += werte[i];
62                  System.out.println("Objekt-Nr. " + i + " Gewicht: " + gewichte[i] +
63                      ↪ " Wert: " + werte[i]);
64              }
65          }
66
67          System.out.println("Gesamtgewicht: " + gesamtGewicht);
68          System.out.println("Gesamtwert: " + gesamtWert);
69      }
70
71      public static void main(String[] args) {
72          werteAus();
73      }

```

Code-Beispiel auf Github ansehen: [src/main/java/org/beschlangaul/examen/examen_66115/jahr_2018/herbst/Rucksack.java](https://github.com/beschlangaul/examen/examen_66115/jahr_2018/herbst/Rucksack.java)