



## DAP2 Übung – Blatt 11

Ausgabe: 4. Juli

Übungsgruppen: Mo. 10. Juli, Mi. 12. Juli, Fr. 14. Juli

Sammelübung: Fr. 14. Juli

**Aufgabe 11.1** (2 Punkte) Gegeben seien die folgenden Punkte in der Ebene:

$$(1, 1), (3, 5), (5, 1), (8, 7), (7, 3), (4, 2), (2, 6)$$

Bestimme mit Hilfe der Spanning-Tree Heuristik eine Lösung für die TSP-Instanz, deren Knoten diese Punkte sind. Die Kosten einer Kante  $(p, q)$  sind gegeben durch die Manhattan-Distanz zwischen den Punkten  $p$  und  $q$ .

*Hinweis:* Die *Manhattan-Distanz* zwischen zwei Punkten  $p = (p_x, p_y)$  und  $q = (q_x, q_y)$  in der Ebene ist:

$$d(p, q) := |p_x - q_x| + |p_y - q_y|$$

**Aufgabe 11.2** (2 Punkte) Wir betrachten die folgenden drei Greedy-Heuristiken für das Rucksackproblem, die sich jeweils nur durch verschiedene Varianten beim Sortieren der Gegenstände unterscheiden:

- (a) Sortiere die Gegenstände nach aufsteigendem Gewicht.
- (b) Sortiere die Gegenstände nach absteigendem Wert.
- (c) Sortiere die Gegenstände nach absteigendem Nutzen (Wert pro Gewicht).

Nach dem Sortieren verfährt jede der Heuristiken wie die in der Vorlesung vorgestellte, d.h. Variante (c) ist der Greedy-Algorithmus aus der Vorlesung.

Gib jeweils eine Beispielinstantz an, bei der Variante (a), (b) und (c) besser ist als die anderen beiden Varianten.

**Aufgabe 11.3** (2 Punkte) Wir betrachten folgende Erweiterung des Greedy-Algorithmus für das Rucksackproblem. (Wir nehmen an, dass  $w_i \leq K$  für alle  $1 \leq i \leq N$ .)

1. Führe zunächst den Greedy-Algorithmus durch. Sei  $z^G$  der Lösungswert der Greedy-Lösung.
2. Sei  $X$  ein Gegenstand mit maximalem Wert  $c_{max} := \max\{c_j \mid 1 \leq j \leq N\}$ .  
Falls  $z^G \geq c_{max}$ , dann nehme die Greedy-Lösung, sonst wähle die Lösung, die nur aus dem Gegenstand  $X$  besteht.

Der Lösungswert ist also  $z^{eG} := \max(z^G, c_{max})$ . Zeige, dass  $z^* \leq 2 \cdot z^{eG}$  gilt, wobei  $z^*$  der Lösungswert der optimalen Lösung ist.

*Hinweis:* Benutze dazu folgende Eigenschaft: Sei  $s$  das erste Element, das der Greedy-Algorithmus nicht wählt, und  $\hat{z} := c_1 + \dots + c_{s-1}$ . Dann gilt  $\hat{z} + c_s \geq z^*$ .

**Aufgabe 11.4** (2 Punkte) Wir betrachten das *Maxsummenproblem*. Gegeben ist eine Folge  $a_1, \dots, a_n$  von reellen Zahlen. Der Wert einer Teilfolge  $a_i, \dots, a_j$  mit  $1 \leq i \leq j \leq n$  ist die Summe  $a_i + \dots + a_j$ . Gesucht ist eine Teilfolge mit maximalem Wert.

Der folgende Algorithmus löst das Maxsummenproblem mit dynamischer Programmierung. Dabei ist  $A_k$  der maximale Wert einer Teilfolge von  $a_1, \dots, a_k$  und  $B_k$  der maximale Wert einer Teilfolge, die mit  $a_k$  endet. Ergänze die freien Stellen im Pseudocode und führe den Algorithmus auf der Eingabe -4, 5, 3, -6, -5, 2, 1, -2, 8, 1, -1 aus.

```

Eingabe: Eine Folge von reellen Zahlen  $a_1, \dots, a_n$  mit  $n \geq 1$ 
Ausgabe:  $a_i, \dots, a_j$  ist eine Teilfolge mit maximalem Wert
 $a_i + \dots + a_j = A_n$ 

 $A_1 := \square$ ;  $i := j := 1$ 
 $B_1 := a_1$ ;  $i' := 1$ 
for  $k := 2$  to  $n$  do
  if  $B_{k-1} \square 0$  then
     $B_k := a_k$ ;  $i' := k$ 
  else
     $B_k := \square$ 
  end if
  if  $B_k > A_{k-1}$  then
     $A_k := \square$ 
     $i := \square$ ;  $j := \square$ 
  end if
end for

```