technische universität dortmund

SoSe 2017

C. Sohler
A. Krivošija, A. Rey, H. Sandvoß

DAP2 – Präsenzübung 5

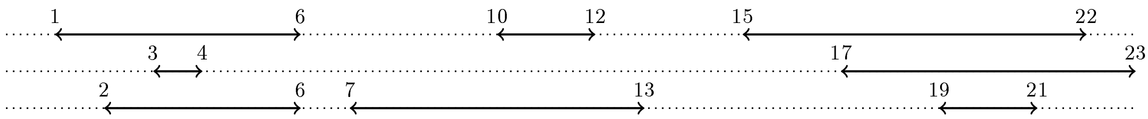
Besprechung: 24.05.2017 — 26.05.2017

Präsenzaufgabe 5.1: (Gierige Algorithmen)

Die Fakultät Informatik bietet einen Help-Desk-Service, der von n Tutorinnen angeboten wird. Die Tutorin T_i, $1 \leq i \leq n$ ist im Dienst im Zeitintervall $[a_i, b_i]$, wobei a_i und b_i positive reelle Zahlen sind. Sei $S = \{[a_1, b_1], \ldots, [a_n, b_n]\}$ die Menge von Intervallen, in denen die Tutorinnen im Dienst sind.

Einige von diesen Tutorinnen sollten als verantwortliche Tutorinnen eingestellt werden, und damit besser bezahlt werden. Sei V die Menge der Dienstzeitintervalle der verantwortlichen Tutorinnen, $V \subseteq S$. Für jede Tutorin T_i mit Dienstzeitintervall $[a, b] \in S$ muss eine verantwortliche Tutorin T_j mit Dienstzeitintervall $[c, d] \in V$ existieren, sodass $[a, b] \subseteq [c, d]$ ist (d.h., es gilt $c \leq a$ und $b \leq d$).

Z.B. stellen für die unten angegebenen Dienstzeitintervalle der n Tutorinnen die Intervalle $V = \{[1, 6], [7, 13], [15, 22], [17, 23]\}$ eine Auswahl der verantwortlichen Tutorinnen dar.



a) Beschreiben Sie einen *gierigen* Algorithmus, der bei Eingabe einer Menge S von Dienstzeitintervallen eine minimale Menge der verantwortlichen Tutorinnen V berechnet (d.h., V enthält minimal viele Intervalle). Geben Sie den Algorithmus auch in Pseudocode an.

b) Analysieren Sie die Laufzeit Ihres Algorithmus.

c) Beweisen Sie, dass Ihr Algorithmus eine optimale Lösung berechnet.

Präsenzaufgabe 5.2: (Gierig ist nicht immer gut)

Zu den bekanntesten und zugleich wichtigsten Optimierungsproblemen zählt das Rundreiseproblem (Traveling Salesman Problem, kurz TSP):

Eingabe: Orte $1, \ldots, n$, sowie die Kosten $c(i, j)$ für $1 \leq i, j \leq n$ mit $i \neq j$, um von i nach j zu reisen.

Ziel: Eine kostenminimale Rundreise, die o.B.d.A. im Ort 1 beginnt und wieder endet und bei der jeder Ort $2 \leq i \leq n$ genau einmal besucht wird. Formal heißt dies, dass eine Permutation π auf $\{1, \ldots, n\}$ mit Fixpunkt $\pi(1) = 1$ gesucht wird, die die Summe

$$c(\pi(n), \pi(1)) + \sum_{i=1}^{n-1} c(\pi(i), \pi(i+1))$$

minimiert.

Gehen Sie davon aus, dass $c(i,j) = c(j,i)$ für alle $i \neq j$ gilt, dass also nur symmetrische Instanzen zugelassen sind, was beim allgemeinen TSP nicht gefordert wird.

a) Welche Laufzeit hat ein Algorithmus, der die kostenminimale Rundreise mittels vollständiger Suche bestimmt, d.h., indem er alle möglichen Rundreisen ausprobiert?

b) Eine gierige Strategie zur Berechnung einer Rundreise würde darin bestehen, dass wir in Ort 1 startend zunächst einen Ort j wählen, für den $c(1,j) = \min\{c(1,k) \mid 2 \leq k \leq n\}$. Von j ausgehend wird dann der nächste Ort l, verschieden von den schon besuchten Orten, so bestimmt, dass $c(j,l)$ unter allen noch nicht besuchten Orten l einen minimalen Wert hat. Dieses Verfahren wird fortgesetzt, bis alle Orte besucht sind.

   Zeigen Sie, dass die gierige Strategie beliebig schlechte Lösungen liefern kann, d.h. zeigen Sie, dass es zu jedem $k > 1$ eine Instanz P für das Rundreiseproblem gibt, sodass

$$\frac{cost_{\text{greedy}}(P)}{cost_{\text{opt}}(P)} > k.$$

   Hierbei bezeichnen $cost_{\text{greedy}}(P)$ die Kosten einer Rundreise, die mit der gierigen Strategie bestimmt wird und $cost_{\text{opt}}(P)$ die Kosten einer optimalen Rundreise.

   Hinweis: Versuchen Sie die gierige Strategie in eine Falle zu locken, von der nur noch teure Kanten zu noch unbesuchten Orten wegführen!