M. Buchin
A. Krivošija, D. Mezlaf
L. Pfahler, A. Rey

DAP2 – Heimübung 6

Ausgabedatum: 11.05.2018 — Abgabedatum: Di. 22.05.2018 bis 12 Uhr

Abgabe:

Schreiben Sie unbedingt immer Ihren vollständigen Namen, Ihre Matrikelnummer und Ihre Gruppennummer auf Ihre Abgaben! Beweise sind nur dort notwendig, wo explizit danach gefragt wird. Eine Begründung der Antwort wird allerdings *immer* verlangt.

Ankündigung:

Der erste Übungstest Test wird am 29.05. während der Vorlesungszeit durchgeführt. Die Bearbeitungsdauer beträgt die kompletten 90 Minuten der Vorlesungsdauer, seien Sie daher bitte zum Test besonders pünktlich, bestenfalls bereits um 12:00 Uhr! Die Verteilung der Studierenden auf die Hörsaale wird nach dem Ablauf der Anmeldefrist (siehe unten) bekannt gegeben.

Für den Test bitten wir zu beachten:

- Als Hilfsmittel ist ausschließlich ein doppelseitig handbeschriebener DIN-A4-Zettel erlaubt. Notieren Sie auf diesem Zettel bitte oben rechts Ihren Namen und Ihre Matrikelnummer.
- Zur Bearbeitung des Tests ist ein dokumentenechter, nicht roter Stift mitzubringen. Nicht erlaubt sind Füllfederhalter oder Tip-Ex.

Um die Durchführung des Tests organisieren zu können, bitten wir um eine Anmeldung zum Test im AsSESS-System. Hierfür ist ein neues Ereignis angelegt, zu dem Sie sich anmelden können, wie Sie es auch schon von den Übungen kennen. Melden Sie sich bitte nur an, wenn Sie wirklich beabsichtigen, am Test teilzunehmen! Die Anmeldung läuft bis Dienstag, 22.05., um 16:00 Uhr.

Aufgabe 6.1 (5 Punkte): (Gierige Algorithmen: ein Scheduling-Problem)

Auf einer Maschine sind m Jobs mit Laufzeiten $p_1, \ldots, p_m$ auszuführen. Wir bezeichnen als *Schedule* für m Jobs eine Reihenfolge, in der diese Jobs auf der Maschine ausgeführt werden. Der Zeitpunkt, zu dem in einem Schedule der Job i beendet ist, sei mit c_i bezeichnet. Gesucht ist ein Schedule, der die summierten Beendigungszeitpunkte von allen Jobs, d. h. $\sum_{i=1}^{m} c_i$ minimiert.

a) Formulieren Sie – sowohl in eigenen Worten als auch in Pseudocode – einen **gierigen** Algorithmus, der bei Eingabe eines Arrays $P[1..m]$, in dem die Laufzeiten p_i, $1 \leq i \leq m$, gespeichert sind, die minimale Summe der Beendigungszeitpunkte für alle Jobs in einem Schedule bestimmt. Für die volle Punktzahl wird ein Algorithmus erwartet, dessen Worst-Case-Laufzeit durch $\mathcal{O}(m \log m)$ beschränkt ist.

b) Analysieren Sie die Laufzeit Ihres Algorithmus.

c) Beweisen Sie, dass dieser Algorithmus den optimalen Wert liefert.

Aufgabe 6.2 (5 Punkte): (Gierige Algorithmen: ein Subset-Sum-Problem)

Eine endliche, natürliche Zahlenfolge $a_1, a_2 \ldots, a_n$ heißt *superwachsend*, falls jede Zahl größer als die Summe der vorherigen Zahlen ist, also

$$a_i > \sum_{j=1}^{i-1} a_j$$

für alle i, $2 \leq i \leq n$, gilt. Es sei $a_1 > 0$.

Wir betrachten nun das folgende Problem: Gegeben sei eine solche superwachsende endliche, natürliche Zahlenfolge und eine natürliche Zahl q, $a_1 \leq q \leq a_n$, für die es eine Teilmenge der Indizes $I \subseteq \{1, 2, \ldots, n\}$ gibt, sodass $\sum_{i \in I} a_i = q$ gilt. Gesucht ist die minimale Größe einer solchen Indexmenge.

a) Beschreiben Sie – mit eigenen Worten und als Pseudocode – einen **gierigen** Algorithmus, der bei Eingabe eines Arrays $A[1..n]$, in dem eine solche Zahlenfolge gespeichert ist, und einer natürlichen Zahl q, $A[1] \leq q \leq A[n]$, die Größe einer minimalen Indexmenge zur Lösung des oben beschriebenen Problems ausgibt. Für die volle Punktzahl wird ein Algorithmus erwartet, dessen Worst-Case-Laufzeit durch $\mathcal{O}(n)$ beschränkt ist.

b) Analysieren Sie die Laufzeit Ihres Algorithmus.

c) Beweisen Sie, dass Ihr Algorithmus die optimale Lösung zurückgibt.