Algorithmen und
Berechnungskomplexität I
Wintersemester 2022/23

UNIVERSITÄT BONN

Prof. Dr. Anne Driemel
Frederik Brüning, Jan Eube
Institut für Informatik

Abgabe: 02.11.2022 bis 10:00 Uhr

Übungsblatt 3

Aufgabe 3.1: Container beladen (6 Punkte)

Ein Unternehmen will n Objekte unterschiedlicher Größen befördern lassen. Dazu kann das Unternehmen beliebig viele Seefracht-Container nutzen, die alle dasselbe Volumen V haben. Die Volumina der Objekte seien durch $v_1, \ldots, v_n$ gegeben und es gelte $v_i \leq V$ für alle $i = 1, \ldots, n$. Im Kontext dieser Aufgabe werden die genauen Abmessungen der Objekte vernachlässigt; d.h. ein Objekt passt in einen Container, wenn das Restvolumen des Containers größer oder gleich dem Volumen des Objektes ist.

Um Transportkosten einzusparen, ist das Unternehmen an einer Aufteilung der Objekte auf möglichst wenige Container interessiert.

(a) Beschreiben Sie einen Greedy-Algorithmus durch Pseudocode, der die Objekte folgendermaßen auf Container aufteilt: In jedem Schritt wird das Objekt, das unter den verbliebenen Objekten das größte Volumen hat, in den ersten passenden Container platziert (d.h., bereits angebrochene Container werden stets in derselben Reihenfolge betrachtet). Sie dürfen zum Sortieren der Eingabe die Methode *sortiere*(A) verwenden, die ein gegebenes Array absteigend sortiert.

(b) Vollziehen Sie Ihren Algorithmus anhand des folgenden Beispiels nach: Die Objekte mit den Volumina $v = (2, 3, 6, 1, 4, 2, 5, 2)$ sollen in Container mit Volumen $V = 6$ einsortiert werden. Begründen Sie, warum die durch Ihren Algorithmus gefundene Lösung in diesem Beispiel optimal ist.

(c) Zeigen Sie anhand eines Gegenbeispiels, dass der Algorithmus nicht für jede Eingabe eine optimale Lösung liefert. Geben Sie einerseits eine optimale Lösung für das beschriebene Beispiel an und andererseits die von Ihrem Algorithmus berechnete Lösung.

Aufgabe 3.2: Bauarbeiter kontrollieren (7 Punkte)

Betrachten Sie die folgende Situation: Sie sind Vorarbeiter(in) auf einer Baustelle und für eine Menge von n Bauarbeitern zuständig. Da auf der Baustelle flexible Arbeitszeiten gelten, gibt es für jeden Arbeiter i einen Zeitpunkt b_i ab dem seine Schicht beginnt und einen Zeitpunkt e_i an dem sie endet. Wir gehen hierbei davon aus, dass der Arbeiter zu dem Start- und Endzeitpunkt jeweils noch auf der Baustelle ist. Sie möchten nun prüfen, ob denn auch jeder Arbeiter gekommen ist. Um dies zu tun, können Sie zu beliebigen Zeitpunkten einmal über die Baustelle gehen und nachsehen, wer denn alles da ist. Der Einfachheit halber lässt sich dies so modellieren, dass dafür keine Zeit benötigt wird, sondern Sie einfach zu diesem Zeitpunkt ohne Verzögerung feststellen können, wer sich auf der Baustelle befindet. Ihr Ziel ist es nun, eine möglichst kleine Menge an Zeitpunkten zu finden, zu denen Sie die die Anwesenheit kontrollieren, sodass Sie jeden Arbeiter mindestens einmal während seiner Schicht gesehen haben.

(a) Geben Sie einen möglichst effizienten Algorithmus (Laufzeit in $O(n \log(n))$) an, welcher für die gegebene Situationen eine möglichst kleine Menge an Kontrollzeitpunkten berechnet.

Hinweis: Modifizieren Sie einen Algorithmus aus der Vorlesung.

(b) Beweisen Sie, dass bei der Ausgabe Ihres Algorithmus tatsächlich alle Bauarbeiter kontrolliert werden.

(c) Analysieren Sie die Laufzeit Ihres Algorithmus.

(d) Beweisen Sie, dass die von Ihnen berechnete Anzahl an Kontrollen tatsächlich minimal ist.

Aufgabe 3.3: Optimale Auswahl von Aufgaben (7 Punkte)

Wir betrachten ein Problem aus dem Bereich *Scheduling*. Die Eingabe besteht wie in der Vorlesung aus einer

Menge $S = \{1, \ldots, n\}$ von Aufgaben. Jede Aufgabe i besitzt einen Startzeitpunkt $s_i \geq 0$ und einen Fertigstellungszeitpunkt $f_i > s_i$. Anders als in der Vorlesung wollen wir aber nicht möglichst viele Aufgaben mit einem Prozessor, sondern alle Aufgaben mit möglichst wenigen Prozessoren bearbeiten. Jeder Prozessor kann zu jedem Zeitpunkt maximal eine Aufgabe bearbeiten.

Zur Lösung des Problems nutzen wir einen Greedy-Algorithmus GREEDY für die Problemvariante aus der Vorlesung mit nur einem Prozessor, auf dem möglichst viele überschneidungsfreie Aufgaben ausgeführt werden sollen. Wir bestimmen mittels GREEDY die Menge der Aufgaben, die vom ersten Prozessor ausgeführt werden. Sind noch Aufgaben übrig, so bestimmen wir von diesen wieder mittels GREEDY eine Teilmenge, die auf dem zweiten Prozessor ausgeführt wird. Dieses Vorgehen wiederholen wir, solange es noch Aufgaben gibt, die nicht zugewiesen sind.

Es stellt sich nun die Frage, welcher Greedy-Algorithmus GREEDY gewählt werden sollte. Bearbeiten Sie hierzu die folgenden Teilaufgaben.

(a) Geben Sie eine Instanz an, für die mit der Methode GREEDYENDE aus der Vorlesung keine optimale Lösung gefunden wird.

(b) Geben Sie einen Greedy-Algorithmus an, mit dem auf jeder Instanz eine optimale Lösung gefunden wird. Zeigen Sie die Korrektheit des Algorithmus.

1. Hinweis: Benutzen Sie einen Algorithmus aus der Vorlesung.

2. Hinweis: Für $x \in \mathbb{R}$ bezeichne $A(x) = \{i \in S : s_i \leq x < f_i\}$ die Menge der Aufgaben i, die zu Zeitpunkt x abgearbeitet werden müssen. Dann ist $\max_{x \in \mathbb{R}} |A(x)|$ eine untere Schranke für die Anzahl an benötigten Prozessoren, da für jedes $x \in \mathbb{R}$ gilt, dass keine zwei Aufgaben aus $A(x)$ mit dem gleichen Prozessor bearbeitet werden können. Nutzen Sie diese untere Schranke um die Korrektheit in Teilaufgabe (b) zu zeigen.