Hans U. Simon
Malte Darnstädt
Thorsten Doliwa

Bochum, den 19.12.2011

Abgabe: 09.01.2012 (bis 15:15 Uhr)

Übungen zur Vorlesung

Diskrete Mathematik

WS 11/12

Übungsblatt 10

Aufgabe 10.1 Um das Maximum und das Minimum einer $n = 2^k$-elementigen Menge zu bestimmen, kann man nach dem Divide-and-Conquer Verfahren rekursiv Minima und Maxima einer Aufteilung der Menge in zwei gleich große Teilmengen bestimmen und das Ergebnis daraus zusammen setzen.

a) Gib den entsprechenden Algorithmus an.

b) Bestimme die Anzahl der Vergleiche zwischen den Elementen, die der Algorithmus ausführt, um Maximum und Minimum einer $n = 2^k$-elementigen Menge zu bestimmen und zeige, dass dies besser als $2n - 2$ wie bei einer naiven Berechnung ist.

Aufgabe 10.2 Eine Hilfsorganisation hat aus weihnachtlichen Spenden n Ladungen mit Lebensmitteln zur Verfügung. Für jede Ladung $i \in [n]$ wurde ein Nährwert $w_i \in \mathbb{N}^+$ und ein Verfallsdatum $d_i \in \mathbb{N}$ ermittelt. Täglich kann nur eine Ladung in das Krisengebiet gebracht werden. Ziel der Helfer ist es, eine Auswahl $F \subset [n]$ von Ladungen zu treffen, so dass keine Ladung $i \in F$ nach ihrem Verfalldatum geliefert wird und der Nährwert aller gelieferten Ladungen maximiert wird.

Zeige, dass dem Problem ein Matroid zugrunde liegt.

Aufgabe 10.3 Gegeben sei ein Rucksack mit Kapazität $B = 10$ und 7 Objekten mit Gewichten $w_1, \ldots, w_7$ und Profiten $p_1, \ldots, p_7$ wobei

i	1	2	3	4	5	6	7
w_i	3	1	4	2	3	2	3
p_i	1	2	4	3	1	3	2

Bestimme den Gesamtprofit einer optimalen Packung des Rucksacks mit Hilfe von dynamischer Programmierung. Gib dazu die vom Algorithmus verwendete Tabelle an.

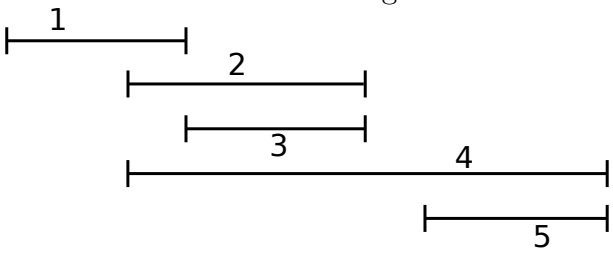
Aufgabe 10.4 Plane deinen Stundenplan für das nächste Semester! Es gibt n Vorlesungen die sich leider teilweise zeitlich überschneiden. Die Vorlesung i beginnt am Zeitpunkt a_i, endet am Zeitpunkt b_i und hat einen Wert von c_i Credit-Points (wobei $i \in \{1, \ldots, n\}$).
Hinweis: Es gilt $a_i, b_i, c_i \in \mathbb{N}$. Falls Vorlesung i genau zu dem Zeitpunkt beginnt, an der Vorlesung j endet (das heißt $a_i = b_j$), ist es möglich beide Vorlesungen zu besuchen.

Gib einen effizienten Algorithmus an, der dynamische Programmierung verwendet, um eine Auswahl von Vorlesungen zu bestimmen, die sich nicht überschneiden und eine maximale Anzahl von Credit-Points liefern.
Nimm dabei an, dass die Vorlesungen nach den Endzeitpunkten b_i sortiert sind und dir eine Funktion $vorher(i)$ zur Verfügung steht, die die Nummer der letzten Vorlesung zurückgibt, die ohne Zeitüberschneidung vor der Vorlesung i besucht werden kann.

Beispiel für 5 Vorlesungen:

i	a_i	b_i	c_i	$vorher(i)$
1	1	4	6	0
2	3	7	3	0
3	4	7	2	1
4	3	11	2	0
5	8	11	6	3

Skizze der Vorlesungszeiten:



Die optimale Belegung bei diesem Beispiel
liefern die Vorlesungen 1, 3 und 5 mit insgesamt 14 Credit-Points.