Höhere Algorithmik, WS 2014/15 — 9. Übungsblatt

20 Betonpunkte. Schriftliche Abgabe in Zweiergruppen bis Montag, 15. Dezember 2014

50. Kürzeste Wege, 0 Punkte

    Berechnen Sie mit dem Bellman/Ford-Algorithmus die kürzesten Wege vom ersten Knoten zu allen anderen Knoten im Graphen mit der folgenden Kostenmatrix:

$$(c_{ij}) = \begin{pmatrix} - & -3 & \infty & 3 & 4 & 2 \\ \infty & - & \infty & \infty & 7 & \infty \\ \infty & -4 & - & 8 & \infty & \infty \\ \infty & \infty & \infty & - & \infty & -3 \\ \infty & \infty & -1 & \infty & - & \infty \\ \infty & \infty & -2 & \infty & \infty & - \end{pmatrix}$$

    Geben Sie die Distanzwerte nach jeder Iteration an.

    Zeichnen Sie die kürzesten Wege im Graphen. (Alle Wege zusammengenommen bilden einen Baum, den *kürzesten-Wege-Baum*.)

51. Inflation, 10 Punkte

    Sie benötigen für die Zukunft n verschiedene Software-Lizenzen, aber wegen beschränkter Mittel können sie nur eine Lizenz pro Monat kaufen. Zu Beginn sind alle Lizenzen gleich teuer, nämlich 1000 Euro. Sie steigen aber jeden Monat im Preis an, nämlich um den Faktor $r_i \geq 1$, $i = 1, \ldots, n$. Die Lizenz für das Produkt i kostet also nach k Monaten $1000r_i^k$ Euro. Sie sollen eine Reihenfolge festlegen, die die Gesamtkosten minimiert.

    Überlegen Sie sich eine passende Greedy-Strategie, und beweisen Sie, dass sie die Optimallösung liefert.

52. Deflation, 0 Punkte

    Funktioniert der Algorithmus der vorigen Aufgabe auch dann, wenn die Ausgangspreise verschieden sind? Beweisen Sie es, oder finden Sie ein Gegenbeispiel.

53. Kritischer Weg, 0 Punkte

    Beim Keksebacken, bei einem Softwareprojekt, oder beim Bau eines Flughafens müssen mehrere Einzelaufgaben durchgeführt werden: den Teig 23 Minuten im Ofen backen; die Zutaten zusammenmischen; den Teig kneten, usw. Für manche Operationen ist eine Reihenfolge vorgegeben: Der Teig kann erst geknetet werden, wenn die Zutaten zusammengemischt sind. Andere Operationen können unahbängig voneinander in beliebiger Reihenfolge oder sogar gleichzeitig durchgeführt werden.

    Gegeben ist also

    - eine Liste $L = \{A_1, A_2, \ldots, A_n\}$ von Aufgaben.
    - Für jede Aufgabe A_i ist die Zeitdauer $t_i > 0$ gegeben, sowie
    - die Menge $F_i \subseteq L$ der Aufgaben, die abgeschlossen sein müssen, bevor A_i beginnen kann. F_i kann auch leer sein.

    Wir nehmen an, dass die Aufgaben schon in einer natürlichen Reihenfolge gegeben sind und dass daher $F_i \subseteq \{A_1, \ldots, A_{i-1}\}$ ist. Wir nehmen auch an, dass genügend Helfer zur Verfügung stehen, sodass es keine Grenze für die Anzahl der parallel durchgeführten Aufgaben gibt. Gesucht ist für jede Aufgabe A_i der frühestmöglichen Beginnzeitpunkt f_i und der frühestmöglichen Endzeitpunkt g_i, bezogen auf den Arbeitsbeginn des Gesamtprojektes.

(a) Schreiben Sie einen Algorithmus in Pseudocode für dieses Problem. Analysieren Sie die Laufzeit und den Speicherbedarf Ihres Algorithmus.

(b) Zeigen Sie, wie das Zeitplanungsproblem als längster Weg von einem Startknoten zu allen anderen Knoten in einem geeigneten Graphen definiert werden kann. Was sind die Knoten dieses Graphen, was sind die Kanten, und welche Gewichte haben die Kanten? Was ist der Startknoten? Was ist die Bedeutung einer Kante (j, k) mit Gewicht c_{jk} in Bezug auf die gesuchten Größen f_i?

54. (10 Punkte) Für welche der unten angegebenen Operationen $\oplus$ und $\otimes$ auf einer Grundmenge S gelten die folgenden beiden Distributivgesetze?

$$a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$$
$$(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$$

Geben Sie Beweise oder Gegenbeispiele an.

(a) $S = \mathbb{R} \cup \{-\infty, \infty\}$, $\oplus = \max$ und $\otimes = \min$.

(b) $S = \mathbb{R}$, $\oplus = \max$ und $\otimes =$ Multiplikation.

(c) $S = \mathbb{R} \cup \{-\infty\}$, $\oplus = +$ und $\otimes = \max$.

(d) $S = \mathbb{R}^2$, $\otimes =$ elementweises $+$: $(a_1, a_2) \otimes (b_1, b_2) = (a_1 + b_1, a_2 + b_2)$, und $\oplus =$ das lexikographische Minimum:

$$(a_1, a_2) \oplus (b_1, b_2) = (\min\{a_1, b_1\}, c), \ \text{ mit } c = \begin{cases} a_2, & \text{für } a_1 < b_1 \\ b_2, & \text{für } b_1 < a_1 \\ \min\{a_2, b_2\}, & \text{für } a_1 = b_1 \end{cases}$$

55. Wahr oder falsch? 0 Punkte

Wenn man alle Kantengewichte eines Graphen (a) quadriert, (b) mit einer Konstanten $C > 0$ multipliziert, (c) zu ihnen eine Konstante $D > 0$ addiert, dann bleibt (i) der kürzeste Weg zwischen zwei Knoten s und t, (ii) der kürzeste Hamiltonsche Kreis, (ii) der kürzeste Hamiltonsche Weg, (iv) der zusammenhängende Teilgraph, der alle Knoten verbindet, mit dem kürzesten Gesamtgewicht, (v) der kürzeste Spannbaum unverändert.

Unterscheiden Sie die Fälle, wo die alle Ausgangsgewichte (α) positiv sind oder (β) beliebig sein können.

56. Eine Ratespiel (zum Knobeln)

Spieler B	Antwort Spielerin A
1	größer als 1
3, 42	zwischen 3 und 42
8, 15, 16, 23, 30	zwischen 8 und 15
9, 10, 11, 12, 13, 14	Es ist 11.

Spielerin A denkt sich eine positive ganze Zahl n aus. Spieler B soll diese Zahl erraten. Er darf in einer Runde eine Liste von Zahlen aufschreiben. Wenn die gesuchte Zahl dabei ist, hat B gewonnen. Andernfalls gibt Spielerin A bekannt, zwischen welchen beiden aufgeschriebenen Zahlen n liegt, bzw. ob sie größer oder kleiner als alle aufgeschriebenen Zahlen ist, siehe nebenstehendes Beispiel. Hauptregel: *Wenn B in einer Runde mehr als n Zahlen aufschreibt, hat er sofort verloren.*

(a) Wie muss die erste Liste von B aussehen? Warum ist die Fragestrategie im obigen Beispiel für B gefährlich?

(b) Finden Sie eine Formel, einen rekursiv definierten Ausdruck, oder ein Berechnungsprogramm (zum Beispiel mit dynamischer Programmierung) für den größten Wert N, sodass alle Zahlen $n \le N$ in höchstens r Runden erraten werden können.

(c) Ändert sich etwas, wenn zu Beginn eine obere Schranke N_0 für n festgelegt wird?