11. Übungsblatt zu Grundzüge von Algorithmen und Datenstrukturen, WS 12/13

Prof. Dr. Markus Bläser, Radu Curticapean, Christian Engels
http://www-cc.cs.uni-sb.de/course/38/

Abgabe: Donnerstag, 31. Januar 2013, 12:00 Uhr

Aufgabe 11.1 In dieser Aufgabe betrachten wir das sogenannte Knapsack-Problem. Gegeben seien eine Gewichtsschranke g und eine Menge von Elementen $\{1, \ldots, n\}$ mit Gewichten $g_1, \ldots, g_n \in \mathbb{Q}_{\geq 0}$ und Profiten $p_1, \ldots, p_n \in \mathbb{Q}_{\geq 0}$. Sie sollen nun eine Teilmenge $S \subseteq \{1, \ldots, n\}$ finden, so dass $p_S := \sum_{i \in S} p_i$ maximiert wird und die Gewichtsschranke nicht überschritten wird, d.h. $g_S := \sum_{i \in S} g_i \leq g$ gilt.

a) *(2 Punkte)* Angenommen wir können Elemente unterteilen. Das bedeutet, wir können für alle $\epsilon \in (0, 1) \subset \mathbb{Q}$ jedes Element i in zwei neue Elemente i_1, i_2 teilen mit $g_{i_1} = \epsilon \cdot g_i$ und $g_{i_2} = (1 - \epsilon) \cdot g_i$, und $p_{i_1} = \epsilon \cdot p_i$ und $p_{i_2} = (1 - \epsilon) \cdot p_i$,

Wir verfolgen nun die folgende Greedy-Strategie: Wir ordnen die Elemente absteigend nach $\frac{p_i}{w_i}$ an und wählen Elemente der Reihe nach, bis das nächste Element k die Gewichtsschranke überschreiten würde. Wir unterteilen k dann so, dass wir unsere Gewichtsschranke g genau erreichen.

Zeigen Sie, dass dieser Greedy-Algorithmus immer eine optimale Lösung findet.

b) *(1 Punkt)* Wenn wir Elemente nicht unterteilen können, ist der oben angegebene Greedy-Algorithmus nicht optimal. Zeigen Sie, dass der Algorithmus ein *beliebig schlechtes* Ergebnis liefern kann.

c) *(1 Punkt)* Wir definieren nun folgenden Algorithmus: Wie oben ordnen wir die Elemente absteigend an und nehmen solange Elemente in die Menge S, bis wir mit dem nächsten Element unsere Gewichtsschranke überschreiten würden. Nun haben wir zwei Mengen: $S = \{1, \ldots, k\}$ (die bisher gewählten Elemente, bei denen die Schranke noch nicht überschritten wird) und $A = \{k + 1\}$ (das Element, bei dem die Schranke überschritten wird). Wir nehmen nun das Maximum $p_{\max}$ aus p_S und p_A. (Wir unterteilen die Elemente nicht.)

Zeigen Sie, dass folgendes gilt: Sei OPT eine optimale Lösung des Knapsack-Problems (ohne teilbare Elemente), dann ist $\frac{1}{2} p_{\text{OPT}} \leq p_{\max} \leq p_{\text{OPT}}$.

Aufgabe 11.2 Sei $G = (V, E)$ ein gerichteter Graph. Zwei Knoten $u, v \in V$ heissen *stark verbunden*, wenn es einen gerichteten Pfad von u nach v gibt und einen gerichteten Pfad von v nach u. (Ein gerichteter Pfad ist eine Folge von Knoten $x_1, \ldots, x_\ell$ mit $\ell \geq 1$, so dass $(x_i, x_{i+1}) \in E$ für alle $1 \leq i < \ell$ gilt.)

a) *(2 Punkte)* Zeigen Sie: Die Relation "stark verbunden" ist eine Äquivalenzrelation.

b) *(2 Punkte)* Die Äquivalenzklassen der Relation "stark verbunden" heißen die *starken Zusammenhangskomponenten* von G. Sei C die Menge der starken Zusammenhangskomponenten. Wir definieren zu G einen gerichteten Graph $H = (C, F)$ wie folgt: Es gibt eine Kante $(c, c') \in F$ genau dann wenn es Knoten $u \in c$ und $u' \in c'$ mit $(u, u') \in E$. Zeigen Sie: Für jeden Graph G ist der so definierte Graph H azyklisch.

Aufgabe 11.3 Betrachten Sie folgenden Algorithmus, der als Eingabe einen gerichteten Graph $G = (V, E)$ erhält.

- Sei S ein leerer Stack.
- Führe DFS-explore auf G aus. Wenn der Zustand eines Knotens auf *finished* gesetzt wird, so speichere den Knoten auf dem Stack.
- Sei $G' = (V, E')$ der Graph, den man erhält, wenn man jede Kante in E umdreht, d.h. $(u, v) \in E'$ genau dann wenn $(v, u) \in E$.
- Führe DFS-explore auf G' aus, wähle dabei die Reihenfolge in der äußeren Schleife gemäß der Reihenfolge in S.

Der Algorithmus berechnet die starken Zusammenhangskomponenten von G.

a) *(3 Punkte)* Zeigen Sie, dass die Knoten, die jeweils während *eines* Aufruf von Depth-First-Search in der Schleife von DFS-explore im letzten Schritt des Algorithmus auf *finished* gesetzt werden, eine starke Zusammenhangskomponente von G bilden.

b) *(1 Punkt)* Was ist die Laufzeit dieses Algorithmus?

Aufgabe 11.4 Sei $G = (V, E, w)$ ein Graph, so dass alle Kantengewichte paarweise verschieden sind, d.h. für alle $e, e' \in E$ mit $e \neq e'$ gilt, dass $w(e) \neq w(e')$. Zeigen Sie: Dann hat G genau einen minimalen Spannbaum.

Aufgabe 11.5 Es sei $G = (V, E)$ ein ungerichteter, einfacher und zusammenhängender Graph mit n Knoten.[1] Wir setzen im Folgenden $V = [n] := \{1, \ldots, n\}$ und bezeichnen den Grad eines Knotens i mit $\deg(i)$.[2] Nun definieren wir zu G eine $(n \times n)$-Matrix $L = L(G)$:

$$L_{i,j} = \begin{cases} \deg(i) & i = j \\ -1 & i \neq j,\ \{i,j\} \in E \\ 0 & i \neq j,\ \{i,j\} \notin E \end{cases}$$

Es sei L' die Matrix, die durch Löschung der n-ten Zeile und der n-ten Spalte aus L entsteht. Zudem sei $\#\mathsf{ST}$ die Anzahl der Spannbäume von G. Wir zeigen im Folgenden

$$\det(L') = \#\mathsf{ST}. \tag{1}$$

Hierfür benötigen wir eine weitere Definition: Es sei H ein beliebiger gerichteter Graph auf n Knoten und m Kanten. Wir bezeichnen die Anzahl der schwachen Zusammenhangskomponenten von H mit $\kappa(H)$. Zu H definieren wir weiterhin eine $(n \times m)$-Matrix $N = N(H)$:

$$N_{v,e} = \begin{cases} +1 & e = (v, *) \\ -1 & e = (*, v)\ . \\ 0 & \text{sonst} \end{cases} \tag{2}$$

a) *(1 Punkt)* Es sei H ein gerichteter Graph auf n Knoten. Zeigen Sie:

$$\operatorname{rank}(N(H)) = n - \kappa(H). \tag{3}$$

b) *(1 Punkt)* Es sei M eine quadratische Matrix, deren Einträge aus $\{-1, 0, +1\}$ stammen. Zudem sei in jeder Spalte von M höchstens ein Vorkommnis von $+1$ und höchstens eines von -1 enthalten. Zeigen Sie, dass dann $\det(B) \in \{-1, 0, +1\}$ gilt.

c) *(2 Punkte)* Es sei A eine $(r \times m)$-Matrix und B eine $(m \times r)$-Matrix, wobei $r \leq m$. Für $S \subseteq [m]$ sei A_{*S} die Matrix, die aus A entsteht, indem alle Spalten gelöscht werden, deren Indizes nicht in S enthalten sind. Ebenso sei B_{S*} die Matrix, die aus B entsteht, indem alle Zeilen gelöscht werden, deren Indizes nicht in S enthalten sind. Zeigen Sie:

$$\det(AB) = \sum_{\substack{S \subseteq [m] \\ |S| = r}} \det(A_{*S}) \det(B_{S*}). \tag{4}$$

d) *(2 Punkte)* Zeigen Sie (1).

e) *(1 Punkt)* Folgern Sie spaßeshalber aus (1), dass es genau n^{n-2} verschiedene beschriftete Bäume auf n Knoten gibt. Die Beschriftungen sind hier aus $\{1, \ldots, n\}$ gewählt, und isomorphe Bäume mit verschiedenen Beschriftungen werden als verschieden gewertet.

f) *(1 Punkt)* Folgern Sie ernsthaft aus (1), dass es einen Algorithmus gibt, der die Anzahl der Spannbäume eines Graphen in Zeit $\mathrm{O}(n^3)$ berechnet.

[1] Ein einfacher Graph besitzt weder Schleifen noch Mehrfachkanten.
[2] Der Grad eines Knotens ist die Anzahl der inzidenten Kanten.