

## Übungsblatt 6

Abgabe der schriftlichen Lösungen bis 3. Juni 2015

**Aufgabe 34** Sei  $G$  ein Graph. **mündlich**

Die *Kantenzusammenhangszahl*  $\lambda(G)$  von  $G$  ist die größte Zahl  $\ell < n$ , so dass  $G - E'$  für jede Menge  $E'$  von  $\ell - 1$  Kanten zusammenhängend ist.  $G$  heißt  $\ell$ -fach kantenzusammenhängend, falls  $\lambda(G) \geq \ell$  ist.

- Zeigen Sie, dass  $\kappa(G) \leq \lambda(G)$  ist.
- Finden Sie einen Algorithmus, der in Linearzeit testet, ob ein Graph  $G$  2-fach kantenzusammenhängend ist.
- Finden Sie einen Algorithmus, der in Zeit  $O(knm)$  testet, ob ein Graph  $G$   $k$ -fach kantenzusammenhängend ist.
- Finden Sie einen  $O(nm \min\{\lambda(G), n^{2/3}\})$  Algorithmus, der  $\lambda(G)$  berechnet. *Bemerkung:* Die Laufzeit kann auf  $O(nm)$  verbessert werden.

**Aufgabe 35** **mündlich**

Finden Sie möglichst effiziente Algorithmen für folgende Probleme. Gegeben ist ein Netzwerk  $N$  und gesucht ist ein (maximaler) Fluss  $f$  mit möglichst

- großem (kleinem) maximalen Kantendurchfluss  $\max_{e \in E} f(e)$ ,
- großem (kleinem) maximalen Knotendurchfluss  $\max_{u \in V \setminus \{s, t\}} f(u)$ ,
- vielen (wenigen) gesättigten Kanten,
- vielen (wenigen) aktiven Kanten  $e$  (d.h.  $f(e) \neq 0$ ).

**Aufgabe 36** **mündlich**

Die  $k$  Arbeitsgruppen eines Betriebs veranstalten ein Geschäftsessen. Um den Informationsaustausch zwischen den einzelnen Gruppen zu fördern, sollen die  $a_i$  Mitglieder jeder Arbeitsgruppe  $i$  so auf die  $r$  zur Verfügung

stehenden Tische verteilt werden, dass an jedem Tisch maximal ein Mitarbeiter aus jeder Gruppe sitzt. Finden Sie einen möglichst effizienten Algorithmus, der eine solche Sitzordnung berechnet (bzw. nachweist, dass es keine gibt), falls Tisch  $j$  maximal  $b_j$  Plätze hat.

**Aufgabe 37** **mündlich**

Für eine  $n$ -köpfige Reisegruppe stehen genau  $n$  Hotelzimmer zur Verfügung. Jede Person  $i$  hat eine bestimmte Vorstellung vom Wert jedes Zimmers  $j$ , die sie in Form von Präferenzwerten  $g(i, j)$  ausdrückt. Dabei soll die Differenz  $g(i, j) - g(i, j')$  in etwa dem Aufpreis entsprechen, den sie für Zimmer  $j$  im Vergleich zu Zimmer  $j'$  zahlen würde.

- Finden Sie einen Algorithmus  $A$ , der jeder Person  $i$  ein Zimmer  $\pi(i)$  mit dem Ziel zuweist, die Zahlungsbereitschaft  $\sum g(i, \pi(i))$  der gesamten Reisegruppe zu maximieren.
- Finden Sie einen Algorithmus  $B$ , der für jedes Zimmer  $j$  einen Preis  $p(j)$  ermittelt, so dass keine Person  $i$  ein anderes Zimmer bevorzugen würde (d.h. für alle  $i, j$  gilt  $g(i, \pi(i)) - p(\pi(i)) \geq g(i, j) - p(j)$ ). *Hinweis:* Verwenden Sie [Aufgabe 38](#) (a).

**Aufgabe 38** **10 Punkte**

- Finden Sie einen Algorithmus  $A$  mit Laufzeit  $O(mn \log n)$ , der für einen bipartiten Graphen  $G = (U, W, E)$  mit Kostenfunktion  $k : E \rightarrow \mathbb{Z}$  für  $i = 1, \dots, \mu(G)$  ein Matching  $M_i$  der Größe  $i$  berechnet, das minimale Kosten unter allen Matchings dieser Größe hat. Zudem soll  $A$  für jedes  $M_i$  eine Preisfunktion  $p$  berechnen, so dass die reduzierten Kosten  $k^p(u, w) = k(u, w) + p(u) - p(w)$  für alle Kanten  $\{u, w\} \in E$  nichtnegativ sind und für  $\{u, w\} \in M_i$  den Wert 0 haben.
- Beschreiben Sie die Arbeitsweise Ihres Algorithmus bei Eingabe des Graphen  $G$  mit folgender Kostenfunktion  $k$ .

