
Universität Karlsruhe (TH)
Institut für Theoretische Informatik

Prof. Dr. Peter Sanders, Dr. Rob van Stee
G. Veit Batz, Johannes Singler

5. Übungsblatt zur Algorithmentechnik WS 2007/08

http://algo2.iti.uni-karlsruhe.de/algotech.php
{sanders|vanstee|batz|singler}@ira.uka.de

Aufgabe 1 (*Streaming / Semi-External Minimum Spanning Tree, Schwierigkeit* $1 + 2$)
Entwerfen Sie einen Algorithmus, der den MST eines Graphen (V, E) berechnet, wobei nicht alle Kanten zur gleichen Zeit in den internen Speicher passen.

a) Nehmen Sie an, dass jede Kante nur einmal zur Verarbeitung vorgelegt wird. Geben Sie einen Algorithmus an, der nichtsdestotrotz den MST mit nur $\mathcal{O}(|V|)$ Speicherplatz berechnet (semi-external: Alle Knoten passen in den internen Speicher, nicht jedoch alle Kanten).

b) Verfeinern Sie Ihren Algorithmus, sodass er bei gleicher Speicherplatzbeschränkung nur noch Laufzeit $\mathcal{O}(|E| \log |V|)$ benötigt.

Aufgabe 2 (*Gewichtetes Matching, Schwierigkeit* $2 + 1$)
Betrachten Sie den Global Path Algorithmus für gewichtetes Matching.

a) Konkretisieren Sie den in der Vorlesung angegebenen Algorithmus so, dass er das Finden der Pfade (ohne Zyklen) in linearer Zeit (in Anzahl der Kanten) erledigt. Erläutern Sie insbesondere die dazu notwendige Datenstruktur.

b) Wie lässt sich die Algorithmus verallgemeinern, dass er auch unter korrekter Berücksichtigung von Zyklen diese Laufzeit einhält?

Aufgabe 3 (*Geldwechselproblem und dynamisches Programmieren, Schwierigkeit* $1 + 3 + 3 + 3$)
Angenommen, Sie haben die Aufgabe, eine Wechselgeldrückgabe für Getränkeautomaten zu programmieren. Ziel ist es dabei, immer die optimale (d. h. kleinstmögliche) Anzahl von Münzen zurückzugeben. Für viele Münzwertsysteme (z. B. auch für den Euro) wird diese Aufgabe durch das Greedy-Verfahren in Abbildung 1 optimal gelöst.

a) Im Fall des Euro ist die Menge der Münzwerte $M = \{1, 2, 5, 10, 20, 50, 100, 200\}$ (alle Werte in Cent). Zeigen Sie: Der Greedy-Algorithmus aus Abbildung 1 liefert nicht immer ein optimales Ergebnis, wenn eine zusätzliche Münze im Wert von 4 Cent eingeführt wird.

Nun soll mit Hilfe von dynamischem Programmieren eine Algorithmus entwickelt werden, der für *jedes beliebige* Münzsystem M mit $1 \in M$ stets eine optimale Lösung liefert (wäre $1 \notin M$ könnten manche Beträge nicht dargestellt werden).

b) Identifizieren Sie die optimalen Teillösungen einer optimalen Lösung. Die Optimalität dieser Teillösungen muss dabei bewiesen werden. Was sind die trivialen Teilprobleme und deren trivial-optimale Lösungen?

c) Geben Sie die optimale Lösung als Rekurrenz an.

d) Geben Sie den fertigen Algorithmus in Pseudo-Code an.

```
1: procedure Geldrueckgabe(rueckbetrag : ℕ0, M : Set of ℕ)
2: rest := rueckbetrag : ℕ0
3: while rest > 0 do
4:    wähle m ∈ M sodass m maximal ist mit m ≤ rest
5:    print Gib eine Münze im Wert von m zurück.
6:    rest := rest − m
7: end while
```

Abbildung 1: Ein Greedy-Verfahren zur Bestimmung der verwendeten Münzen bei der Geldrückgabe. Alle Werte sind natürliche Zahlen und werden z. B. in Cent angegeben. Die Menge M umfasst die möglichen Münzwerte.

Aufgabe 4 (*Devisenhandel und kürzeste Wege mit allgemeinen Kantengewichten, Schwierigkeit* 2)
Gegeben sei eine endliche Menge von Währungen $W = \{w_1, \ldots, w_k\}$. Die Wechselkurse zwischen den einzelnen Währungen seien durch eine Matrix $U = ((u_{ij})) \in \mathbb{R}_{\geq 0}^{k \times k}$ gegeben: Für eine Einheit der Währung w_i erhält man u_{ij} Einheiten der Währung w_j (entsprechend gelte $u_{ii} = 1$ für alle i). Gewinnbringender Devisenhandel ist nun offenbar dann möglich, wenn es ein Folge von Umtauschgeschäften gibt, die mit *einer* Einheit einer Währung $w \in W$ startet und mit mehr als einer Einheit derselben Währung endet.
Wie kann man feststellen, ob eine gegebene Matrix von Wechselkursen gewinnbringenden Devisenhandel erlaubt? Hinweis: Verwenden Sie den Zusammenhang $\log(xy) = \log x + \log y$ und betrachten Sie kürzeste Wege in Gegenwart von allgemein-reellen Kantengewichten.