



Algorithmen und Datenstrukturen

Sommersemester 2022

Übungsblatt 10

Abgabe: Dienstag, 2. Juli, 2024, 10:00 Uhr

Aufgabe 1: Dijkstra Lowerbound

(10 Punkte)

Wir haben in der Vorlesung gesehen, dass die Laufzeit von Dijkstra mit Fibonacci heaps $O(m+n \log n)$ ist. Doch ist dies die echte Laufzeit von Dijkstra, oder ist unsere Analyse nur nicht gut genug, oder vielleicht unser Heap nicht gut genug? Um diese Frage zu beantworten zeigen wir das die Abhängigkeit in m und die Abhängigkeit in n beide korrekt sind.

- Argumentieren Sie, dass jeder Algorithmus für SSSP (Single Source Shortest Paths) mindestens $\Omega(m)$ Zeit braucht. Hier reicht eine Intuitive Beschreibung. (1 Punkt)
- Beweisen Sie, dass Dijkstra die kürzesten Wege in sortierter Reihenfolge ausgibt. Für einen Startknoten v wird also für alle anderen Knoten $u \neq w$ die Distanz $d(v, u)$ vor $d(v, w)$ ausgegeben, wenn $d(v, u) < d(v, w)$. Wir erwarten eine genaue Beschreibung und saubere formale Argumente. (4 Punkte)
- Beweisen Sie, dass Dijkstra $\Omega(n \log n)$ Zeit benötigt, wenn der Algorithmus mit einem Vergleichsbasierten Heap implementiert wird. Die Idee ist folgende: Reduzieren Sie das Sortieren von n Zahlen auf das Problem kürzeste Wege auszurechnen. (Also gegeben n Zahlen in einem Array kreieren Sie eine Instanz von SSSP.) Wir erwarten eine genaue Beschreibung und saubere formale Argumente. (5 Punkte)

Aufgabe 2: Währungsarbitrage

(10 Punkte)

Gegeben seien n Währungen w_1, \dots, w_n . Die Umrechnungskurse der Währungen seien in einer $n \times n$ -Matrix A mit Einträgen a_{ij} ($i, j \in \{1, \dots, n\}$) gegeben. Eintrag a_{ij} ist der Umrechnungskurs von w_i nach w_j , d.h. für eine Einheit von w_i bekommt man a_{ij} Einheiten von w_j .

Gegeben eine Währung w_{i_0} möchte man herausfinden, ob es eine Folge i_0, i_1, \dots, i_k gibt, sodass man Gewinn macht, wenn man eine Einheit von w_{i_0} zu w_{i_1} tauscht, danach zu w_{i_2} etc. bis zu w_{i_k} und schließlich wieder zurück zu w_{i_0} .

- Formulieren Sie die Fragestellung als Graphproblem. Definieren Sie dazu einen geeigneten Graphen sowie eine Bedingung, welche der Graph genau dann erfüllt, wenn es eine Folge von Währungen wie oben beschrieben gibt. (4 Punkte)
- Geben Sie einen Algorithmus an, welcher in $\mathcal{O}(n^3)$ Zeitschritten entscheidet, ob es eine Folge von Währungen wie oben beschrieben gibt. Begründen Sie Laufzeit und Korrektheit. (6 Punkte)

Hinweis: Es gilt $a \cdot b > 1 \iff (-\log a) + (-\log b) < 0$.