Algorithmen für Routenplanung

Übung 4

Thomas Pajor

Lehrstuhl für Algorithmik I
Institut für theoretische Informatik
Karlsruher Institut für Technologie
Universität Karlsruhe (TH)

22. Juni 2009

Aufgabe 1: Contraction Hierarchies

Gegeben: Graph $G = (V, E, \text{len})$.

Aufgabe 1: Contraction Hierarchies

Gegeben: Graph $G = (V, E, \text{len})$.

(a) Wie lässt sich die 1-Hop-Zeugensuche bei der Knotenreduktion möglichst schnell berechnen?

Hinweis: Benutzen Sie Ideen des Many-to-Many Routing-Ansatzes.

Aufgabe 1: Contraction Hierarchies

Gegeben: Graph $G = (V, E, \text{len})$.

(a) Wie lässt sich die 1-Hop-Zeugensuche bei der Knotenreduktion möglichst schnell berechnen?

(b) Geben Sie ein effizientes Verfahren für die 2-Hop-Zeugensuche an, das ohne einer lokalen DIJKSTRA-Suche auskommt.

Hinweis: Benutzen Sie Ideen des Many-to-Many Routing-Ansatzes.

Aufgabe 2: Transit-Node Routing

Gegeben: Graph $G = (V, E, \text{len})$ und $L + 1$ Level von Transit-Knoten

$$T_L \subseteq \cdots \subseteq T_1 \subseteq T_0 := V.$$

Aufgabe 2: Transit-Node Routing

Gegeben: Graph $G = (V, E, \text{len})$ und $L + 1$ Level von Transit-Knoten

$$T_L \subseteq \cdots \subseteq T_1 \subseteq T_0 := V.$$

(a) Beschreiben Sie eine effiziente Vorgehensweise zur Bestimmung der Access-Nodes $\overrightarrow{A}_l(v)$ für alle Level $l \leq L$.

Hinweis: Benutzen Sie einen Top-Down-Ansatz und beziehen Sie bei der Level-l-Berechnung Zwischenergebnisse aus höheren Level $> l$ mit ein.

Aufgabe 2: Transit-Node Routing

Gegeben: Graph $G = (V, E, \text{len})$ und $L + 1$ Level von Transit-Knoten

$$T_L \subseteq \cdots \subseteq T_1 \subseteq T_0 := V.$$

(a) Beschreiben Sie eine effiziente Vorgehensweise zur Bestimmung der Access-Nodes $\overrightarrow{A}_l(v)$ für alle Level $l \leq L$.

(b) Wie lassen sich Distanztabellen D_l für alle $l \leq L$ effizient berechnen?

Hinweis: Benutzen Sie einen Top-Down-Ansatz und beziehen Sie bei der Level-l-Berechnung Zwischenergebnisse aus höheren Level $> l$ mit ein.