Algorithmen für Routenplanung

Übung 3

Thomas Pajor

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

8. Juni 2009

Aufgabe 1: Arc-Flags, Aufwärmübung

Gegeben: Graph $G = (V, E, \mathrm{len})$ und eine Partition $\mathcal{P} := \{C_1, \ldots, C_k\}$

Aufgabe 1: Arc-Flags, Aufwärmübung

Gegeben: Graph $G = (V, E, \mathrm{len})$ und eine Partition $\mathcal{P} := \{C_1, \ldots, C_k\}$

(a) Sei $|\mathcal{P}| = 1$.
Es werde die All-Pair-Shortest-Path Variante zur Vorberechnung benutzt. Unterscheidet sich der Suchraum ($\#$ rel. Kanten und $\#$ abgearb. Knoten) des Arc-Flags-Algorithmus von DIJKSTRA's Algorithmus?

Aufgabe 1: Arc-Flags, Aufwärmübung

Gegeben: Graph $G = (V, E, \text{len})$ und eine Partition $\mathcal{P} := \{C_1, \dots, C_k\}$

(a) Sei $|\mathcal{P}| = 1$.
Es werde die All-Pair-Shortest-Path Variante zur Vorberechnung benutzt. Unterscheidet sich der Suchraum (# rel. Kanten und # abgearb. Knoten) des Arc-Flags-Algorithmus von DIJKSTRA's Algorithmus?

(b) Sei $|\mathcal{P}| = n$.
Zeigen Sie: Eine s-t-Anfrage besucht ausschließlich Knoten entlang des kürzesten Weges. Muss dieser dazu eindeutig sein?

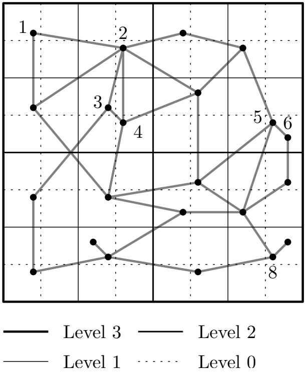
Aufgabe 2: Multi-Level Paritionierung

Gegeben sei ein Graph $G = (V, E)$. Definiere ML-Partition $\mathcal{P}$ mit k Levels wie folgt:

- $\mathcal{P}_k := \{V\}$
- Für $i < k$:

$$\mathcal{P}_i := \bigcup_{C \in \mathcal{P}_{i+1}} P_C,$$

  wobei P_C eine Partition der Zelle C ist.
- comlev(u, v) ist das kleinste Level i für das ein $C \in \mathcal{P}_i$ existiert mit $u, v \in C$.

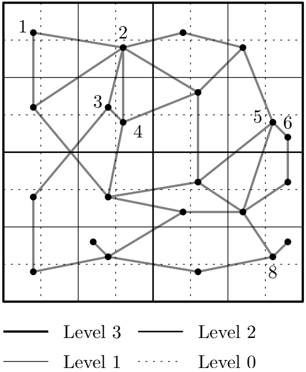


Aufgabe 2: Multi-Level Paritionierung

Gegeben sei ein Graph $G = (V, E)$. Definiere ML-Partition $\mathcal{P}$ mit k Levels wie folgt:

Aufgabe:

(a) Geben Sie comlev für die Knotenpaare $(1, 2)$, $(1, 4)$, $(5, 6)$ und $(3, 8)$ an.



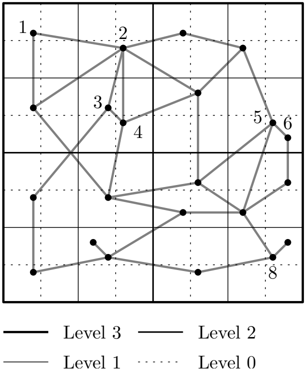
Aufgabe 2: Multi-Level Paritionierung

Gegeben sei ein Graph $G = (V, E)$. Definiere ML-Partition $\mathcal{P}$ mit k Levels wie folgt:

Aufgabe:

(a) Geben Sie comlev für die Knotenpaare $(1, 2)$, $(1, 4)$, $(5, 6)$ und $(3, 8)$ an.

(b) Geben Sie ein effizientes Verfahren zur Berechnung von comlev an.



Aufgabe 3: Multi-Level Arc-Flags

Gegeben: Graph $G = (V, E)$, multi-level Partition $\mathcal{P} = \{\mathcal{P}_0, \ldots, \mathcal{P}_k\}$.

Aufgabe 3: Multi-Level Arc-Flags

Gegeben: Graph $G = (V, E)$, multi-level Partition $\mathcal{P} = \{\mathcal{P}_0, \ldots, \mathcal{P}_k\}$.

(a) Zeigen Sie: Der gemeinsame Level $\mathrm{comlev}(v_i, t)$ entlang Knoten v_i eines kürzesten Weges nimmt nicht notwendigerweise monoton ab.

Aufgabe 3: Multi-Level Arc-Flags

Gegeben: Graph $G = (V, E)$, multi-level Partition $\mathcal{P} = \{\mathcal{P}_0, \ldots, \mathcal{P}_k\}$.

(a) Zeigen Sie: Der gemeinsame Level $\text{comlev}(v_i, t)$ entlang Knoten v_i eines kürzesten Weges nimmt nicht notwendigerweise monoton ab.

(b) Welche Konsequenz ergibt sich für die Vorberechnung auf Basis von Randknoten?

Aufgabe 3: Multi-Level Arc-Flags

Gegeben: Graph $G = (V, E)$, multi-level Partition $\mathcal{P} = \{\mathcal{P}_0, \ldots, \mathcal{P}_k\}$.

(a) Zeigen Sie: Der gemeinsame Level $\text{comlev}(v_i, t)$ entlang Knoten v_i eines kürzesten Weges nimmt nicht notwendigerweise monoton ab.

(b) Welche Konsequenz ergibt sich für die Vorberechnung auf Basis von Randknoten?

(c) Geben Sie einen Algorithmus in Pseudocode zur Vorberechnung an.

Aufgabe 3: Multi-Level Arc-Flags

Gegeben: Graph $G = (V, E)$, multi-level Partition $\mathcal{P} = \{\mathcal{P}_0, \ldots, \mathcal{P}_k\}$.

(a) Zeigen Sie: Der gemeinsame Level $\mathrm{comlev}(v_i, t)$ entlang Knoten v_i eines kürzesten Weges nimmt nicht notwendigerweise monoton ab.

(b) Welche Konsequenz ergibt sich für die Vorberechnung auf Basis von Randknoten?

(c) Geben Sie einen Algorithmus in Pseudocode zur Vorberechnung an.

(d) Erweitern Sie DIJKSTRA's Algorithmus zu einer ML-Arc-Flags-Query.

Aufgabe 3: Multi-Level Arc-Flags

Gegeben: Graph $G = (V, E)$, multi-level Partition $\mathcal{P} = \{\mathcal{P}_0, \dots, \mathcal{P}_k\}$.

(a) Zeigen Sie: Der gemeinsame Level $\text{comlev}(v_i, t)$ entlang Knoten v_i eines kürzesten Weges nimmt nicht notwendigerweise monoton ab.

(b) Welche Konsequenz ergibt sich für die Vorberechnung auf Basis von Randknoten?

(c) Geben Sie einen Algorithmus in Pseudocode zur Vorberechnung an.

(d) Erweitern Sie DIJKSTRA's Algorithmus zu einer ML-Arc-Flags-Query.

(e) Zeigen Sie die Korrektheit Ihres Algorithmus aus (d) wenn Sie die Vorberechnung aus Aufgabe (c) benutzen.

Aufgabe 4: Self-Bounding Bid. Reach-Algorithmus

Gegegeben: Graph $G = (V, E)$, Reach-Werte $r(v)$ für alle Knoten.

Aufgabe 4: Self-Bounding Bid. Reach-Algorithmus

Gegegeben: Graph $G = (V, E)$, Reach-Werte $r(v)$ für alle Knoten.

(a) Zeigen Sie, dass der kürzeste Weg unter Benutzung des Abbruchkriteriums des normalen Bi-DIJKSTRA Algorithmus nicht notwendigerweise im Suchraum enthalten ist.

Aufgabe 4: Self-Bounding Bid. Reach-Algorithmus

Gegegeben: Graph $G = (V, E)$, Reach-Werte $r(v)$ für alle Knoten.

(a) Zeigen Sie, dass der kürzeste Weg unter Benutzung des Abbruchkriteriums des normalen Bi-Dijkstra Algorithmus nicht notwendigerweise im Suchraum enthalten ist.

(b) Zeigen Sie die Korrektheit des Abbruchkriteriums aus der Vorlesung:
- Stoppe eine Suche sobald minKey$(Q) \geq \mu/2$ gilt.
- Sind beide Suchen gestoppt, gib kürzesten Weg μ aus.