Algorithmen und Datenstrukturen

Übungsblatt 10

Ioan Oleksii Kelier Florian Keppeler

Aufgabe 1

(a) Um den kürzesten Weh zu bestimmen, muss man jede Kante überprüfen. Daraus folgt, dass jeder Algorithmus für SSSP mindestens $\Omega(m)$ Zeit braucht.

(b) Diese Eigenschaft ergibt sich direkt aus der Tatsache, dass wir im Dijkstra-Algorithmus eine Prioritätswarteschlange verwenden.

Angenommen, wir haben zwei Elemente u und w aus der Prioritätswarteschlange entfernt. Es folgt, dass $d(v, u) < d(v, w)$ gilt, da u vor w aus der Prioritätswarteschlange entfernt wurde.

Mit anderen Worten: Diese Eigenschaft ist gegeben, weil die Knoten in aufsteigender Reihenfolge ihrer Distanz in die Prioritätswarteschlange aufgenommen werden, wodurch gewährleistet ist, dass jeder nachfolgende Knoten größer ist als der vorherige.

(c) Angenommen, wir wollen n Zahlen $S := \{\lambda_1..., \lambda_n \mid \lambda_i \in \mathbb{R}_{\geq 0}\}$ sortieren. Sei $G = (V, E),\ V := \{v_0\} \cup S,\ \ E := \{(v_0, v_i)|v_i \in V\}$ ein Graph mit Gewichten, die durch die folgende Funktion dargestellt werden

$$w : E \to S,\ w(v_0, v_i) \mapsto \lambda_i \text{ für } i \in \{1, ..., n\}$$

(Es gilt $|V| = n + 1$ wegen v_0 mit $v_0.d = 0$, da wir irgendwo anfangen müssen). Aus dem Teil (b) wissen wir, dass wir als Ergebnis eine sortierte Folge von Zahlen erhalten, die als kürzester Weg dargestellt wird.

Da *Insert* und *Extract-Min* $O(log(n))$ Zeit kosten, und wir wiederum *2n (n Einfügen + n Extrahieren-Min)* Operationen durchführen, folgt daraus, dass die Zeit $O(nlog(n))$ ist. Da aber diese Sortierung auf einem Vergleich basiert, folgt daraus, dass diese Version von Dijkstra-Algorithmus $\Omega(nlog(n))$ Zeit benötigt.

(Thats a tricky part, I'm somewhat confused about jumping from BigO to Omega, despite it's actually makes sence, since this art of sorting is bounden from below by Omega(nlogn))