



Universität Freiburg
Institut für Informatik
PD Dr. A. Heinz
PD Dr. S. Schuierer

Georges-Köhler-Allee, Geb. 051
D-79110 Freiburg i. Br.
Tel. (0761) 203-8164
Tel. (0761) 203-8165
Freiburg, 27. November 2000

Algorithmentheorie Übungsblatt 4

Abgabe bis *Montag*, 11.12.2000, 15 Uhr, (in der Vorlesung oder den Kästen
“Algorithmentheorie” im Geb. 051, EG)
Besprechung am *Mittwoch*, 13.12.2000, 11-13 Uhr

Aufgabe 1: (2 + 2 (+ 1) Punkte) Kürzeste Wege

Sei $G = (V, E)$ ein gerichteter Graph mit Kostenfunktion $c : E \rightarrow \mathbb{R}^+$ und $h : V \rightarrow \mathbb{R}^+$ eine Gewichtung der Knoten von G . Wir definieren eine Neugewichtung $\hat{c} : E \rightarrow \mathbb{R}$ der Kanten von G durch

$$\hat{c}(u, v) = c(u, v) - h(u) + h(v).$$

- Zeigen Sie, daß ein Weg W genau dann ein kürzester Weg von u nach v bzgl. der Kostenfunktion c ist, wenn W auch ein kürzester Weg von u nach v bzgl. der Kostenfunktion \hat{c} ist.
- Für alle Kanten $e = (u, v)$ gelte: $h(u) \leq h(v) + c(u, v)$. Der Algorithmus von Dijkstra werde auf den mit \hat{c} gewichteten Graphen angewandt. Zudem werde s als erstes Element von R mit Entfernung $h(s)$ zu sich selbst eingefügt. Was läßt sich über die berechnete Distanz eines Knotens t mit $h(t) = 0$ sagen? Wie sieht die berechnete Distanz im allgemeinen Fall aus?
- (Nur für Absolventen der Vorlesung “Grundlagen der künstlichen Intelligenz”) Welche Beziehung besteht zwischen der oben skizzierten Variante des Algorithmus’ von Dijkstra und dem A^* -Verfahren, falls die Heuristik h in dem A^* -Verfahren die Ungleichung $h(u) \leq h(v) + c(u, v)$ erfüllt?

Aufgabe 2: (4 Punkte) Union-Find Strukturen

Zeigen Sie, daß jede Folge von m *make-set*-, *find-set*- und *union*-Operationen, von denen n *make-set*-Operationen sind, höchstens $O(m \log n)$ viele Schritte benötigt, falls die Strategie *Vereinigung nach Höhe* angewandt wird, bei der immer die Wurzel des höheren der beiden Bäume zum Vater der Wurzel des anderen Baumes wird.

Aufgabe 3: (4 Punkte) Minimale Spannbäume

In der Vorlesung wurde eine effiziente Implementierung von Kruskals Verfahren zur Berechnung minimal spannender Bäume angegeben. Dabei wurden die Datenstrukturen Priority

Queue und Union-Find-Strukturen verwendet. Geben Sie auch für das andere besprochene Verfahren zur Berechnung minimal spannender B-aume, den Algorithmus von Prim, eine effziente Implementierung an und analysieren Sie deren Laufzeit.

Aufgabe 4: (4 Punkte) Huffman-Codes

Verallgemeinern Sie den Algorithmus von Huffman auf ternäre Zeichencodes, d.h. Codeworte, die die Symbole 0, 1 und 2 verwenden. Formulieren Sie die Lemmas neu, mit deren Hilfe man die Korrektheit Ihres Algorithmus' zeigen kann. (Die Lemmas müssen nicht bewiesen werden.)