Algorithmen und Programmierung 3, WS 2003/2004 — 11. Übungsblatt

Abgabe bis Donnerstag, 5. Februar 2004

---

62. (0 Punkte) Wie kann man auf einfache Art überprüfen, ob ein Graph, der mit Adjazenzlisten gespeichert ist, ein *einfacher Graph* ist (keine mehrfachen Kanten enthält)? Versuchen Sie, mit $O(m+n)$ Zeit und mit möglichst wenig zusätzlichem Speicher auszukommen.

63. (0 Punkte) Wenden die den Algorithms von Dijkstra zur Bestimmung der kürzesten Wege im folgenden gerichteten Graphen an: $V = \{1, 2, \ldots, 6\}$, der Graph ist vollständig, das heißt, von jedem Knoten gibt es eine Kante zu jedem anderen Knoten, und die Kantenlängen sind $c_{ij} = 3^{j-i}$ für $i < j$ und $c_{ij} = i - j$ für $i > j$. Der Startknoten ist der Knoten 2.

64. (6 Punkte) Konstruieren Sie einen Graphen, der auch Kanten mit negativer Länge enthält, und bei dem der Algorithmus von Dijkstra die kürzesten Wege nicht richtig bestimmt. Der Graph soll keine Kreise negativer Länge enthalten.

65. (0 Punkte) Denken Sie sich einen gierigen Algorithmus aus, der versucht, in einem Graphen einen kurzen Weg von einem Startknoten s zu einem Zielknoten t zu finden. Welche Problem können dabei auftreten?

66. (9 Punkte) Betrachten Sie die Variante einer Halde, bei der jeder innere Knoten $d \geq 2$ Kinder hat. (Für $d = 2$ ergeben sich die gewöhnlichen Halden aus der Vorlesung.) Wie verändert sich die Laufzeit für die Methoden `zugroß` beziehungsweise `zuklein` in Abhängigkeit von d? Wie wirkt sich das auf die Operationen *einfügen*, *entferneMin* und *verkleinereSchlüssel* aus?

    Wenn man eine solche „d-Halde“ für den Algorithmus von Dijkstra verwendet, wie muss man dann d in Abhängigkeit von m und n wählen, dass man die optimale asymptotische Laufzeit erhält? Welche Laufzeit ergibt sich dann für das kürzeste-Wege-Problem? Was ergibt sich in den Extremfällen $m = \Theta(n)$ und $m = \Theta(n^2)$?

67. (0 Punkte) Zeigen Sie, dass der folgende *Algorithmus von Prim* einen kürzesten Spannbaum findet:

    $G := \{s\};\ \mathit{Abstand}[s] := 0;$
    $N := V - \{s\};$
    while $G \neq \emptyset$ **do**
        entferne einen Knoten v mit kleinstem $\mathit{Abstand}[v]$ aus G;
        füge v in B ein;
        für alle Kanten $(v, w) \in E$, die von v ausgehen, **do**
            if $w \in N$ **or** $(w \in G$ **and** $\mathit{Abstand}[w] > c_{vw})$
            then $G := G \cup \{w\};\ N := N - \{w\};$
                    $\mathit{Abstand}[w] := c_{vw};$
                    $\mathit{Vorgänger}[w] := v;$

    Was ist der Unterschied dieses Algorithms zum Algorithmus von Dijkstra für kürzeste Wege?

68. (a) (0 Punkte) Berechnen Sie die Verschiebefunktion des *Fibonacci-Wortes*

    $$abaababaabaababaababaabaababaabaab.$$

    (b) (5 Punkte) Berechnen Sie die Verschiebefunktion der *Morse-Folge*

    $$0110100110010110100101100110100110010110011010010110100110010110.$$

    (0 Punkte) Wie gehen die beiden Folgen weiter?

69. (freiwillige Zusatzaufgabe, 0 Punkte) Wie kann man n Brüche, deren Zähler und Nenner zwischen 1 und n liegen, in linearer Zeit sortieren?