Algorithmen, Datenstrukturen, Datenabstraktion, 2019/20 — Klausur

zu bearbeiten bis Freitag, 26. Juni 2020, 11:45 Uhr

1. Graphenalgorithmen, 10 Punkte

   Die sogenannte *Erdős-Zahl* ist nach dem berühmten ungarischen Mathematiker Paul Erdős (1913–1996) benannt. Er selbst hat Erdős-Zahl 0. Alle seine Koautorinnen und Koautoren, mit denen er wissenschaftliche Arbeiten publiziert hat, haben Erdős-Zahl 1. Wer eine gemeinsame Arbeit mit jemandem mit Erdős-Zahl 1 geschrieben hat, aber nicht mit Erdős selbst, hat Erdős-Zahl 2, und so weiter. Personen, die nicht auf diese Weise mit Erdős verbunden sind, haben Erdős-Zahl ∞.

   Der *Kollaborationsgraph* enthält alle lebenden oder toten Personen als Knoten. Zwei Knoten sind durch eine Kante verbunden, wenn die beiden Personen eine gemeinsame wissenschaftliche Arbeit publiziert haben.

   (a) Schreiben Sie einen *Algorithmus*, der für eine gegebene Person im Kollaborationsgraphen die Erdős-Zahl berechnet.

   (b) Schreiben Sie einen Algorithmus, der die größte endliche Erdős-Zahl bestimmt.

   Algorithmen aus der Vorlesung können Sie direkt verwenden; die müssen Sie nicht "ausprogrammieren".

2. Digitale Suchbäume, 10 Punkte

   Zeichne sie (a) den digitalen Suchbaum, (b) den komprimierten digitalen Suchbaum für die Bitketten `0011000101`, `001111001`, `01011`, `00111111`, `01010`, `001101`. Zeigen Sie, wie die Bäume nach dem Einfügen des Wertes `0011100` aussehen. Entfernen Sie danach `00111111` und zeichnen Sie die Bäume wieder.

3. Algebraischen Spezifikation für Mengen, 10 Punkte

   Die Operationen *einfüge*, *lösche*, *leereMenge*, und *istenthalten* für Mengen von ganzen Zahlen sind folgendermaßen spezifiziert (wie in der Vorlesung):

$$\begin{array}{lrll}
(1) & \mathit{istenthalten}(x, \mathit{leereMenge}) & = \mathit{False} & \\
(2) & \mathit{istenthalten}(x, \mathit{einfüge}(x, m)) & = \mathit{True} & \\
(3) & \mathit{istenthalten}(x, \mathit{einfüge}(y, m)) & = \mathit{istenthalten}(x, m), & \text{für } x \neq y \\
(4) & \mathit{istenthalten}(x, \mathit{lösche}(x, m)) & = \mathit{False} & \\
(5) & \mathit{istenthalten}(x, \mathit{lösche}(y, m)) & = \mathit{istenthalten}(x, m), & \text{für } x \neq y
\end{array}$$

   Leiten Sie durch Umformungen und Fallunterscheidungen folgende Identität her, und geben Sie bei jedem Schritt die Nummer der Gleichung an, die Sie verwenden.

$$\mathit{istenthalten}(u, \mathit{lösche}(x, \mathit{einfüge}(x, m))) = \mathit{istenthalten}(u, \mathit{lösche}(x, m))$$

4. Dynamisches Programmieren, 10 Punkte

   Gegeben ist eine Folge $(a_0, a_1, a_2, \ldots, a_{n-1})$ von n verschiedenen Zahlen, z. B. (5, 28, 19, 15, 20, 33, 12, 17, 10). Eine *absteigende Teilfolge* ist eine Teilfolge, die monoton fallend ist, zum Beispiel (19, 12, 10) oder die Teilfolge (20). (Jede einelementige Teilfolge ist eine absteigende Teilfolge.) Schreiben Sie ein effizientes Programm in Pseudocode, das die *längste* absteigende Teilfolge berechnet (nicht nur deren Länge).

   Analysieren Sie die Laufzeit und den Speicherbedarf Ihres Algorithmus.

   Geben Sie eine kurze Begründung für die Korrektheit an.

   Hinweis: Es geht mit $O(n)$ Speicher und $O(n^2)$ Zeit, aber auch (komplizierter) mit $O(n \log n)$ Zeit.