

Algorithmen und Komplexität Übungsblatt 8

Diese Serie simuliert eine Prüfung und alle Aufgaben werden mit Punkten bewertet. Nutzen Sie diese Möglichkeit um herauszufinden wieviele Punkte Ihre Lösungen an einer Prüfung erhalten würden.

Peer-Grading: Aufgabe 2. Bewerten Sie die Lösung anhand des vorgegebenen Punkteschemas in der Musterlösung.

Korrektur durch die Assistenten: Aufgaben 1 und 3

* * *

Aufgabe 1

Im Algorithmus von Kruskal ist in jedem Schritt zu testen, ob eine Kante $e = \{u, v\}$ in dem bereits konstruierten Wald einen Kreis schliesst. Erklären Sie, wie man eine Union-Find Datenstruktur einsetzen kann, um Kruskal effizient zu implementieren. Zeigen Sie, dass die Implementierung mit der Union-Find Datenstruktur denselben Spannbaum generiert, wie das „naiven“ Verfahren, welches mit Breitensuche überprüft, ob die Knoten u und v in der selben Zusammenhangskomponenten liegen und e somit einen Kreis schliessen würde. Vergleichen Sie die Laufzeit des „naiven“ Verfahren mit der Laufzeit der Implementierung mit der Union-Find Datenstruktur. Welcher Effizienzvorteil ergibt sich im Vergleich zu diesem „naiven“ Testverfahren? (10 Punkte)

Aufgabe 2

Wir betrachten ein zweidimensionales Array A , bestehend aus n Zeilen und m Spalten. Jedes Element im Array enthält eine natürliche Zahl. Wir nehmen an, dass keine Zahl mehrfach enthalten ist. Zwei Elemente im Array sind benachbart, wenn sie sich an einer Seite berühren. Innere Elemente haben also vier Nachbarn, Randelemente entsprechend drei oder zwei Nachbarn.

Eine Folge x_1, x_2, \dots, x_k von Elementen im Array heisst aufsteigende Sequenz, wenn für jedes $i \in \{1, \dots, k-1\}$ die Elemente x_i und x_{i+1} im Array benachbart sind und jeweils $x_i < x_{i+1}$ gilt.

- a) Finden Sie in untenstehendem Array eine längste aufsteigende Sequenz.

12	5	2	10
4	6	9	3
14	11	8	1

(2 Punkte)

- b) Benutzen Sie das Prinzip der dynamischen Programmierung um einen Algorithmus mit Laufzeit $\mathcal{O}(mn \log(nm))$ zu entwerfen, welcher in einem gegebenen $n \times m$ -Array die längste aufsteigende Teilsequenz findet. Beschreiben Sie Ihren Algorithmus, analysieren Sie die Laufzeit und beweisen Sie die Korrektheit. (14 Punkte)

Aufgabe 3

In der Vorlesung haben Sie (a, b) -Bäume kennengelernt, welche die Operationen FIND, INSERT und DELETE unterstützen. Erweitern Sie die Datenstruktur der (a, b) -Bäume, sodass die Operation $\text{SELECT}(k, T)$ in Zeit $\mathcal{O}(\log n)$ ausgeführt werden kann. Die Operation $\text{SELECT}(k, T)$ soll das Element mit dem k -kleinsten Schlüssel des (a, b) -Baumes finden. Erklären Sie auch, wie die Operationen FIND, INSERT und DELETE angepasst werden müssen, und begründen Sie dabei auch, dass die Laufzeit dieser Operationen nach wie $\mathcal{O}(\log n)$ beträgt. (12 Punkte)

ABGABE DER HAUSAUFGABEN IN DER VORLESUNG AM 12.11.2019.