

AuD Übung – Blatt 7

Ausgabedatum: 21.01.2010 — Abgabedatum: 27.01.2010

Aufgabe 7.1: Randomisierte Suchbäume

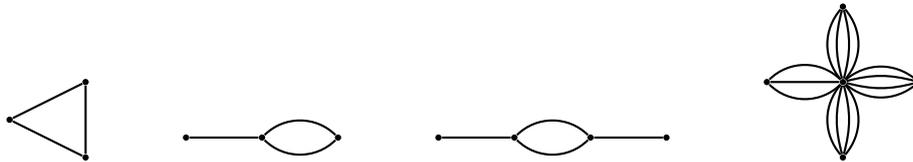
Sei $S = \{s_1, \dots, s_n\}$ eine Menge von Elementen mit $s_k < s_\ell$ für $k < l$ und sei außerdem für jedes Element $s_\ell \in S$ eine Priorität $prio(s_\ell)$ mit $prio(s_k) \neq prio(s_\ell)$ für $k \neq \ell$ gegeben. Zeige die folgende Aussage: Im Treap T für S das Element s_j genau dann Vorgänger von s_i , wenn gilt:

$$prio(s_j) = \min\{prio(s_\ell) \mid s_\ell \in S_{i,j}\},$$

wobei $S_{i,j} = \{s_i, s_{i+1}, \dots, s_j\}$ für $i < j$ und $S_{i,j} = \{s_j, s_{j+1}, \dots, s_i\}$ für $i > j$ ist.

Aufgabe 7.2: Randomisierter MinCut-Algorithmus

Berechne für die folgenden Beispielgraphen die (exakte) Erfolgswahrscheinlichkeit für den *einfachen* randomisierten MinCut-Algorithmus aus der Vorlesung.



Aufgabe 7.3: Verbessern der Erfolgswahrscheinlichkeit

Der einfache randomisierte MinCut-Algorithmus hat eine Erfolgswahrscheinlichkeit von $\frac{1}{n^2}$ bzw. eine Fehlerwahrscheinlichkeit von $1 - \frac{1}{n^2}$. Senke die Fehlerwahrscheinlichkeit auf $\frac{1}{k}$ (die Erfolgswahrscheinlichkeit soll also mindestens $1 - \frac{1}{k}$ sein). Welche Laufzeit erhältst Du für den verbesserten Algorithmus?

Aufgabe 7.4: Randomisierter MinCut-Algorithmus

Erkläre, wie man den einfachen randomisierten MinCut-Algorithmus in Zeit $O(n^2)$ implementieren kann. Gehe dabei darauf ein, wie man unter den verbleibenden Kanten eine Kante gleichverteilt und zufällig auswählt.