Einführung in die Diskrete Mathematik
4. Übung

1. Sei T ein kostenminimaler aufspannender Baum für einen ungerichteten Graphen G mit nichtnegativen Kantengewichten. G' entstehe aus G, indem ein neuer Knoten s hinzugefügt wird, der mit jedem Knoten aus $V(G)$ durch eine (ebenfalls gewichtete) Kante verbunden ist. Zeigen Sie, wie man aus T und G' in linearer Laufzeit einen kostenminimalen aufspannenden Baum für G' berechnen kann. (5 Punkte)

2. Ein Telekommunikationsnetzwerk werde durch einen ungerichteten Graphen $G = (V, E)$ modelliert, dessen Kanten voneinander unabhängige Ausfallwahrscheinlichkeiten $p : E \to [0, 1]$ haben. Wie findet man in Zeit $O(m + n \log n)$ einen spannenden Baum, der die Wahrscheinlichkeit, dass alle seine Kanten funktionieren, maximiert? (4 Punkte)

3. a) Zeigen Sie, dass es Folgen von Heap-Operationen gibt, so dass in einem Fibonacci-Heap die maximale Pfadlänge in einer Arboreszenz $\Theta(n)$ ist, wenn n die Zahl der Elemente ist.
   b) Zeigen Sie, dass zwei Fibonacci-Heaps mit n_1 und n_2 Elementen in $O(\log(n_1 + n_2))$ Zeit verschmolzen werden können. Das Ergebnis soll also ein Fibonacci-Heap sein, der alle n_1+n_2 Elemente enthält. (3+2 Punkte)

4. Wir wollen eine Menge von Objekten verwalten. Jedem Objekt ist ein Schlüssel zugeordnet, und keine zwei Objekte haben denselben Schlüssel. Außerdem gibt es eine lineare Ordnung auf allen Schlüsseln. Insbesondere wollen wir effizient ein Element mit einem gegebenen Schlüssel suchen. Mit einem sortierten Array geht das durch binäre Suche offenbar in logarithmischer Zeit. Zusätzlich wollen wir aber auch effizient neue Elemente einfügen, was bei einem sortierten Array im allgemeinen lineare Laufzeit erfordern würde. Betrachten Sie statt dessen die folgende Datenstruktur:

   Wir speichern die Objekte in einer Menge von sortierten Arrays, die alle unterschiedliche Länge haben. Die Länge eines jeden Arrays soll eine Zweierpotenz einer ganzen Zahl sein. Jedes Element der Menge soll in genau einem Array vorkommen.

   (a) Zeigen Sie, dass es stets eine solche Zerlegung in Arrays gibt und dass die Längen der Arrays eindeutig sind.
   (b) Beschreiben Sie, wie man ein Element mit gegebenem Schlüssel in Zeit $O(\log^2 n)$ finden kann, wenn n die Zahl der eingefügten Elemente ist.
   (c) Zeigen Sie, dass man, wenn man mit einer leeren Menge beginnt, für eine Folge von n Einfügeoperationen Zeit $O(n \log n)$ benötigt. (2+2+2 Punkte)

Abgabe: Donnerstag, den 17.11.2016, vor der Vorlesung.