ETH Zürich
Institut für Theoretische Informatik
Prof. Dr. Angelika Steger
Florian Meier, Ralph Keusch

HS 2016
Abgabe 01.11.2016

Algorithmen und Komplexität
Übungsblatt 6

Aufgabe 1

Es sei ein Graph $G = (V, E)$ mit einer Gewichtsfunktion $w : E \longrightarrow \mathbb{N}$ gegeben, und es sei ein minimaler Spannbaum T von G gegeben. Wir verringern das Gewicht $w(e)$ einer Kante e, die nicht in T enthalten ist, und nennen den so modifizierten Graphen G'. Geben Sie einen Algorithmus an, der ausgehend von T in Laufzeit $\mathcal{O}(|V|)$ einen minimalen Spannbaum von G' findet.

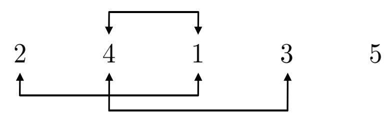
Aufgabe 2

Gegeben sei ein Array $a[1 \ldots 2n]$ mit $2n$ Zahlen. Beschreiben Sie in Pseudocode einen Algorithmus, der als Input n Paare von Zahlen $((x_1, y_1), \ldots, (x_n, y_n))$ bekommt, wobei $1 \leq x_i \leq n \leq y_i \leq 2n$, und der in Laufzeit $\mathcal{O}(n)$ ein Array $b[1 \ldots n]$ berechnet, wobei $b[i]$ die kleinsten Zahl der Menge $\{a[x_i], a[x_i + 1], \ldots, a[y_i]\}$ sei.

Aufgabe 3

Sei $(a_1, a_2, \ldots, a_n)$ eine Liste von paarweise unterschiedlichen Zahlen aus $\{1, 2, \ldots, n\}$. Das Paar (i, j) ist eine *Inversion* falls $i < j$ und $a_i > a_j$. Das Ziel ist die Anzahl der Inversionen in der Liste zu berechnen.

Diese Anzahl misst, wie weit die Liste $(a_1, \ldots, a_n)$ von der sortierten Liste $(1, 2, \ldots, n)$ entfernt ist. Zum Beispiel gibt es in der folgenden Liste 3 Inversionen: $(1, 3)$, $(2, 3)$ und $(2, 4)$.



(a) Wie gross ist die Anzahl Inversionen in einer Liste $(a_1, \ldots, a_n)$ minimal und maximal? Geben Sie für beide Fälle je ein Beispiel.

(b) Beschreiben Sie (kurz und in Worten) einen einfachen Algorithmus, der die Inversionen zählt und in Zeit $O(n^2)$ läuft. Begründen Sie die Laufzeit.

Wir entwerfen einen schnelleren Algorithmus, der die Anzahl Inversionen in Zeit $O(n \log n)$ berechnet. Dazu soll der Mergesort Algorithmus verwendet werden.

(c) Zuerst betrachten wir eine Liste L in der die Elemente in der linken und der rechten Hälfte jeweils aufsteigend sortiert sind. Wie kann man den Merge-Schritt von Mergesort so erweitern, dass er aus den beiden Hälften eine sortierte Liste erstellt und gleichzeitig die Anzahl der Inversionen von L zählt? Achten Sie darauf, dass der Merge-Schritt weiterhin in Zeit $O(n)$ läuft.

(d) Verwenden Sie (c), um den Mergesort Algorithmus so zu erweitern, dass er die Anzahl der Inversionen einer beliebigen Liste zählt. Der Algorithmus soll in Zeit $O(n \log n)$ laufen. Begründen Sie die Korrektheit und Laufzeit des Algorithmus.

Abgabe der Hausaufgaben in der Vorlesung am 01.11.2016.