Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Prof. Dr. Susanne Albers
Dr. Alexander Souza

SS 2007
Übungsblatt 3

Randomisierte Algorithmen und probabilistische Methoden

Aufgabe 1 Minimaler Schnitt

Gegeben sei ein Graph $G = (V, E)$ mit n Knoten und m Kanten. Ein *Schnitt* in G ist eine Teilmenge $S \subseteq V$, so dass weder $S = \emptyset$ noch $S = V$ gilt. Die Menge $E(S) = \{\{u, v\} \in E : u \in S, v \in V - S\}$ bezeichnet die Kanten *über* den Schnitt S; $e(S) = |E(S)|$ deren Anzahl.
Die Aufgabe besteht darin einen minimalen Schnitt S^* in G zu finden, d.h., einen mit minimaler Kantenzahl $e(S^*)$. Betrachten Sie folgenden Algorithmus.
Solange G mehr als zwei Knoten hat wiederhole:

(i) Ziehe eine Kante e zufällig und gleichverteilt.

(ii) Kontrahiere e zu einem neuen Knoten. Dabei entstehende Mehrfachkanten werden übernommen; Schlingen werden gelöscht.

Die (Mehrfach-)Kanten die der resultierende Graph hat geben den berechneten Schnitt an.
Sei S^* ein fester minimaler Schnitt in G.

(1) Zeigen Sie, dass die Wahrscheinlichkeit, dass obiger Algorithmus S^* findet mindestens $\frac{2}{n(n-1)}$ ist?
Hinweis. Betrachten Sie dazu die Ereignisse E_i mit

$$E_i \text{ tritt ein falls die } i\text{-te kontrahierte Kante } \textit{nicht} \text{ aus } S^* \text{ ist.}$$

Berechnen Sie die bedingten Wahrscheinlichkeiten $\Pr[E_i \mid E_1 \cap \cdots \cap E_{i-1}]$ und damit die Erfolgswahrscheinlichkeit $\Pr[E_1 \cap \cdots \cap E_{n-2}]$.
Erinnern Sie sich an $\Pr[A \cap B] = \Pr[A \mid B] \Pr[B]$.

(2) Wie oft muss man den Algorithmus wiederholen, damit er mit Wahrscheinlichkeit mindestens $\frac{1}{2}$ einen minimalen Schnitt in G findet?

Aufgabe 2 Bubblesort

Sei $x_1, \ldots, x_n$ eine Liste von n unterschiedlichen Zahlen. Die Zahlen x_i und x_j heißen *invertiert*, falls $i < j$ und $x_i > x_j$. Der Algorithmus Bubblesort vertauscht solange benachbarte invertierte Zahlen bis es keine mehr gibt (und die Liste daher in aufsteigender Reihenfolge sortiert ist).
Angenommen die Eingabe für Bubblesort ist eine zufällig gleichverteilt gezogene Permutation von $1, 2, \ldots, n$. Was ist die erwartete Anzahl von Vertauschungen die der Algorithmus durchführt?