Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Institut für Theoretische Informatik
Emo Welzl, Jiří Matoušek

22.04.2004

Theoretische Informatik (Kernfach) *SS 2004*
Exercise Set 5

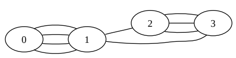
Erinnerung:
Am 30.04. findet die Semesterzwischenprüfung statt. Studenten eingeschrieben in die Übungsgruppen 1–5 legen die Prüfung in NO C 3, Punkt 9:00 bis Punkt 10:00 ab, Studenten eingeschrieben in die Übungsgruppen 6,7 in HG G 3, Punkt 10:00 bis Punkt 11:00.

Exercise 1

An undirected multigraph $G = (V, E)$ is a graph where two (distinct) vertices can be connected by more than one edge, i.e. E is a multiset of two-element sets (no loops). Such a graph can be represented by its adjacency matrix $A_G \in \mathbb{N}^{V \times V}$ where

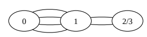
$$A_G(u, v) = \text{number of edges between } u \text{ and } v.$$

Given an edge $\{u, v\}$ in G a contraction of this edge results in a new graph G' with vertex set $V' = V \setminus \{u, v\} \cup \{\boxed{uv}\}$, where $\boxed{uv}$ is a new vertex. The edges in G not connected to u or v survive in G', the edges between u and v vanish and edges from a third vertex w to u or v become edges $\{w, \boxed{uv}\}$.

For example the graph 

has adjacency matrix

$$\begin{pmatrix} 0 & 4 & 0 & 0 \\ 4 & 0 & 1 & 1 \\ 0 & 1 & 0 & 3 \\ 0 & 1 & 3 & 0 \end{pmatrix}$$

and if one contracts edge $\{2, 3\}$ the resulting graph is 

.

Assume for an algorithm on such multigraphs we need three operations:

1. Determine the number of edges between two given vertices.
2. Select an edge of G uniformly at random.
3. Contract a given edge of G.

As the algorithm will use all three operations quite frequently we want to store G in an efficient way.

Describe a data structure together with algorithms implementing these three operations. The goal is to perform the first operation in time $O(1)$ and the other two operations in time $O(n)$. (Here n is the number of vertices of the current G. Note that by contracting an edge the number of vertices decreases.)

Exercise 2

Prove Proposition 1.19 in the lecture notes from Hall's theorem.

Exercise 3

Let (G, s, t, c) be a network with all capacities equal to 1. Assume that there is no s-t-cut with capacity k or smaller ($k > 1$ integer). Prove that there are directed s-t-paths P_1, P_2,..., P_k such that P_i and P_j have no common edges for $1 \leq i < j \leq k$. (Two paths may have common vertices.)

Hint: Consider a suitable maximum flow and extract the paths by induction.

Exercise 4

For $i, n \in \mathbb{N}, 1 \leq i \leq n$, let $R_n^{(i)}$ be the depth of the key of rank i in a *rightist* random search tree for n keys. (Recall the definition from Exercise 1.19, p. 12 in the hand-out *Random(ized) Search Trees.*)

Determine $E[2^{R_n^{(n)}}]$.

Exercise 5

For a random search tree on n nodes and $i, j, k \in \{1, \ldots n\}, i \leq j$, define the random indicator variable

$$C_{i,j}^k = [\text{node } k \text{ is common ancestor of nodes } i \text{ and } j]$$

(where "node k" stands for "node holding key of rank k", etc.).

Determine $\Pr[C_{i,j}^k = 1]$ for $i, j, k \in \{1, \ldots n\}$, $i \leq j$.

Hint: You'll have to discriminate the cases $k < i \leq j$, $i \leq k \leq j$, and $i \leq j < k$.

Due-date:29.04.04 in the lecture