ETH 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

02.06.2005

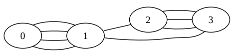
Theoretische Informatik (Kernfach) — *SS 2005*
Exercise Set 8

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 $A_G \in \mathbf{N}^{V \times V}$ where

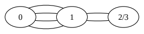
$$A_G(u, v) = \text{number of edges between } u \text{ and } v.$$

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

A graph G has vertex set $\{1, 2, \ldots, 4n\}$, and vertices i and j are connected by an edge if

(i) $1 \le i, j \le 2n$, or

(ii) $2n + 1 \le i, j \le 4n$, or

(iii) $i \le n$ and $j = i + 2n$.

What is the size of a minimum cut in G? Prove your answer!

If you fail for general n try to solve the exercise for $n = 3$.

Exercise 3

For a graph G a vertex cut is a set C of vertices such that removing the vertices in C and the incident edges makes the graph disconnected. Note that a complete graph has no vertex cut. Let $\kappa_e(G)$ be the size of a minimum edge cut in G and $\kappa_v(G)$ the size of a minimum vertex cut. (Define for the complete graph on n vertices $\kappa_v(K_n) := n - 1$.)

(a) Prove that $\kappa_e(G) \geq \kappa_v(G)$.

(b) Does there exist a function $f : \mathbf{N} \to \mathbf{N}$, such that for any graph G the inequality $\kappa_e(G) \leq f(\kappa_v(G))$ holds?

Exercise 4

We consider the following algorithm BETTERMINCUT for minimum cut, with an input graph G on n vertices and with additional parameters t and T:

```
BetterMinCut(G, t, T):
  H ← RandomContract(G, t)
  return minimum of
      T runs of BasicMinCut(H)
```

```
RandomContract(G, t):
  while |V(G)| > t do
      for random e ∈ E(G)
          G ← G/e
  end while
  return G
```

```
BasicMinCut(G):
  while G has more than 2 vertices do
      pick a random edge e in G
      G ← G/e
  end while
  return the size of the only cut in G
```

In the following, you are asked to analyze the behavior of this algorithm for various settings of t and T. The goal is to make really (!) rough calculations to realize what is going on. You need not worry about integer parts, say. For the probability of success of RANDOMCONTRACT(G, t) use the bound derived in class, which is roughly t^2/n^2, $t \geq 2$.

(a) Consider BETTERMINCUT called with $t = n/2$ and $T = 2$. How do the probability of success and the running time change compared to the basic guessing algorithm BASICMINCUT(G)?

(b) Consider BETTERMINCUT called with $t = \sqrt{n}$ and $T = n$. Estimate the probability of success. How many times do we need to repeat this algorithm, in order to make the probability of success at least $\frac{1}{2}$? What is the total running time of these repetitions?

(c) Consider now $t = n^\alpha$, $T = n^\beta$, where $\alpha \in (0, 1)$ and $\beta > 0$ are constants. Again estimate the success probability and number N of repetitions needed to make the success probability at least $\frac{1}{2}$. What choice of α and β give the best total running time? (If you cannot determine the very best ones, at least give the best ones you can find.)

Exercise 5 (Challenge)

Prove that no graph on n vertices has more than $\binom{n}{2}$ minimum cuts.
Hint: What happens if Karger's algorithm avoids contracting edges from a given minimum cut?

Due-date: 09.06.05 in the lecture