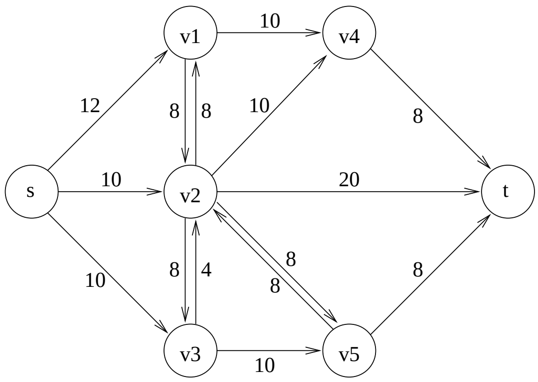
Technische Universität München
Department of Computer Science
Chair for Efficient Algorithms
Prof. Dr. Sami Khuri

WS 1999/2000
Problem Set 3
May 16, 2000

# Efficient Algorithms and Data Structures II

## Problem 1 (8 credits)

Use the preflow-push algorithm to produce a maximum flow for the following network.




## Problem 2 (12 credits)

The *edge connectivity* of an undirected graph is the minimum number $k$ of edges that must be removed to disconnect the graph. For example, the edge connectivity of a tree is 1, and the edge connectivity of a cyclic chain of vertices is 2. Show how the edge connectivity of an undirected graph $G = (V, E)$ can be determined by running a maximum-flow algorithm on at most $|V|$ flow networks, each having $O(V)$ vertices and $O(E)$ edges.

## Problem 3 (20 credits)

Let $G = (V, E)$ be a flow network with source $s$ and sink $t$, and suppose that each edge $e \in E$ is restricted to capacity $c(e) = 1$. Assume also, for convenience, that $|E| = \Omega(V)$.

a. Suppose we run the basic Ford-Fulkerson maximum-flow algorithm on $G$. What is the worst-case running time? Justify briefly.

   (Hint: the total running time of Ford-Fulkerson is at most $O(E \cdot |f^*|)$ in general, where $f^*$ is the maximum flow found by the algorithm. But here, with the stated restriction, we have a tighter bound.)

b. Suppose a maximum flow for $G$ has been computed, and a new edge with capacity 1 is added to $E$. Describe how the maximum flow can be efficiently updated. Analyze your algorithm.

   (Hint: it is not only the value of the flow that must be updated, but the flow itself.)

c. Suppose a maximum flow for $G$ has been computed, but an edge is now removed from $E$. Describe how the maximum flow can be efficiently updated. Analyze your algorithm.