Assignment 6

Post Date: 23 Nov 2015    Due Date: 30 Nov 2015, 1 pm
You are permitted and encouraged to work in groups of two.

Problem 1: Minimum Cut 6 Points

(a) Find a minimum cut for the following graph with the algorithm of Stoer and Wagner. Start with vertex $a$ and comment each step of the algorithm.

(b) Which conclusion in the proof of correctness of the algorithm of Stoer and Wagner requires nonnegative edge weights? Find an example graph with negative edge weights such that the algorithm of Stoer and Wagner does not find a minimum cut.

Problem 2: Vertex Capacities 4 Points

Let $((V, E); s, t; c)$ be an extended flow network where not only edge capacities, but also vertex capacities are constrained, i.e., $c : E \cup V \rightarrow \mathbb{R}_0^+$ and a flow $f : E \rightarrow \mathbb{R}_0^+$ must satisfy, in addition to the edge capacity constraint and the flow conservation constraint, the following vertex capacity constraint

$$\sum_{w: (w, v) \in E} f(w, v) \leq c(v) \text{ for all } v \in V.$$

Show that determining a maximum flow in a network with additional vertex capacities can be reduced to determining a maximum flow in a network with only edge capacities.
Problem 3: Fibonacci Heaps – Cascading Cut  

In the following we consider the Fibonacci heap from the lecture, but without cascading cut.

Let $T_k(i)$ be a Fibonacci heap with $k$ root nodes with 0 to $k$ children, where the root node with $i$ children is missing. No tree has nodes of depth two or more.

![Figure 1: Illustration of a Fibonacci heap $T_6(4)$, where the root node with 4 children is missing.](image)

(a) Given a heap $T_k(i)$, show how to construct $T_k(i - 1)$ using $O(k)$ Insert, ExtractMin, and DecreaseKey (without cascading cut) operations.

(b) Given a heap $T_k(0)$, show how to construct $T_{k+1}(k + 1)$ in constant time.

(c) Conclude that a heap $T_k(k)$ can be constructed with $O(k^3)$ operations.

(d) Show that you can construct a $T_k(k)$ from a $T_k(k)$ by applying 2 Insert and 2 ExtractMin operations that need $\Omega(k)$ time.

(e) Conclude that the worst-case runtime is in $\Omega(m^{1+\epsilon})$ for $m$ operations of Insert, ExtractMin, and DecreaseKey without cascading cuts for some $\epsilon > 0$. 