COMP 362 - Winter 2015 - Assignment 1

Due: 6pm Jan 30th.

General rules: In solving these questions you may consult books but you may not consult with each other. You should drop your solutions in the assignment drop-off box located in the Trottier Building.

1. (10 Points) For every integer $n \geq 2$, construct a flow network on n vertices such that every s, t-cut (A, B) is a minimum cut. In your construction all the capacities must be strictly positive, and the underlying graph must be connected. How many minimum cuts does this network have?

2. (20 Points) Consider the following two algorithms for finding the maximum flow:

Algorithm 1: Scaling max-flow
• Initally set $f(e) := 0$ for all edges e.
• Set Δ to be $\max c_e$ rounded down to a power of 2.
• While $\Delta \geq 1$:
•   While there is an s, t-path P in $G_f(\Delta)$:
•   Augment the flow using P and update $G_f(\Delta)$.
•   Endwhile.
•   Set $\Delta := \Delta/2$.
• Endwhile.
• Output f.

Algorithm 2: The fattest path algorithm

- Initally set $f(e) := 0$ for all edges e.
- While there exists an s,t-path in G_f:
- Augment the flow using the fattest s,t-path P in G_f.
- Update G_f.
- Endwhile.
- Output f.

In the second algorithm the fattest means the largest bottleneck. From the class we know that the number of augmentations in Algorithm 1 is at most $2m\lceil\log_2 K\rceil$, where K is the maximum capacity of an edge. Deduce from this that the number of augmentations in Algorithm 2 is also at most $2m\lceil\log_2 K\rceil$.

3. (15 points) Consider the variant of the maximum flow problem where every node v also has an integer capacity $c_v \geq 0$. We are interested in finding the maximum flow as before, but now with the extra restriction that $f^{\text{in}}(v) \leq c_v$ for every node v. Solve this problem using the original maximum flow problem.

4. (a) (10 points) Show that for every flow network, there exists an execution of the Ford-Fulkerson algorithm that never decreases the value of the flow on any of the edges (i.e. never "pushes back" the flow on any of the edges).

   (b) (5 points) On the other hand, show that if we modify the Ford-Fulkerson algorithm so that it does not decrease the flow on any of the edges (i.e. we do not add the opposite edges to the residual graph), then the algorithm might terminate without finding the maximum flow.

5. (20 points) Let $(G, s, t, \{c_e\})$ be a flow network, and let F be the set of all edges e that belong to some minimum cut. That is, there exists at least one minimum cut (A, B) such that e goes from A to B. Give a polynomial time algorithm that finds all edges in F. What is the running time of your algorithm.

6. (20 points) Is it true that for every flow network with at most m edges, there exist a sequence of at most m augmentations that lead to a maximum flow?