Massachusetts Institute of Technology
6.854J/18.415J: Advanced Algorithms
Ankur Moitra

Wednesday, March 2, 2016

Problem Set 4

Due: Wednesday, March 9, 2016 – 7 pm
Dropbox Outside Stata G5

Collaboration policy: collaboration is *strongly encouraged*. However, remember that

1. You must write up your own solutions, independently.
2. You must record the name of every collaborator.
3. You must actually participate in solving all the problems. This is difficult in very large groups, so you should keep your collaboration groups limited to 3 or 4 people in a given week.
4. Write each problem in a separate sheet and write down your name on top of every sheet.
5. **No bibles. This includes solutions posted to problems in previous years.**

1 Prove, or Disprove?

Which of the following statements about flows are true and which are false? Justify your answer with a (short) proof or counterexample.

(a) In any maximum flow, and for all vertices v and w, either the amount of flow from v to w ($f(v, w)$) or the amount of flow from w to v ($f(w, v)$) is 0.

(b) Consider a directed graph $G = (V, E)$. There always exists a maximum flow of G such that, for all vertices v and w, either the amount flow on vw or the amount of flow on wv is 0.

(c) If all directed edges in a network have distinct capacities, then there is a unique maximum flow.

(d) If we replace each directed edge in a network with two directed edges in opposite directions with the same capacity and connecting the same vertices, then the value of the maximum flow remains unchanged.

(e) If we add the same positive number λ to the capacity of every directed edge, then the minimum cut (but not its value) remains unchanged.

(f) Flow is transitive: if in a graph there is a flow of value v from s to t, and there is a flow of value v from t to u, then there is a flow of value v from s to u.

2 Dynamic Max Flow Problem

Suppose you have already computed the maximum flow in a network with m edges and integral capacities using an augmenting-paths algorithm.

(a) Show how to update the maximum flow in $O(m)$ time after increasing a specified capacity by 1.

(b) Show how to update the maximum flow in $O(m)$ time after decreasing a specified (positive) capacity by 1.

3 Critical Edges for Max Flow

(a) An edge is *upward critical* if increasing its capacity increases the maximum flow value. Does every network have an upward-critical edge? Give an algorithm to identify all upward-critical edges in a network. Its running time should be substantially better than that of solving m maximum-flow problems.

(b) An edge is *downward critical* if decreasing its capacity (by any amount) decreases the maximum flow value. Is the set of upward-critical edges the same as the set of downward-critical edges? Describe an algorithm for identifying all downward-critical edges, and analyze your algorithm's worst-case complexity.

4 Lunchtime

At lunchtime it is crucial for people to get to the food trucks as quickly as possible. We will show how to use maximum flow to solve this problem. Consider the following model. The building is represented by a graph $G = (V, E)$, where each room, landing, or other location is represented by a vertex and each corridor or stairway is represented by an edge. Each corridor has an associated capacity c, meaning that at most c people can pass through the corridor at once. Traversing a corridor from one end to the other takes one timestep. (Traversing a room takes zero time.)

(a) Suppose all people are initially in a single room s, and that the building has a single exit t. Show how to use maximum flow to find a fastest way to get everyone out of the building. Hint: create an auxiliary graph G' that has vertices to represent each room *at each time step*.

(b) Show that the same technique can be used when people are initially in multiple locations and there are multiple exits.

(c) Generalize the approach to where different corridors have different (integral) transit times.

(d) **(Optional)** Suppose transit times are not integral. Is there still a way to solve the problem?

(e) **(Optional)** The above algorithm is polynomial in the number of people. Can you improve it to be polynomial (in the graph size) regardless of the number of people?

5 Capacity Scaling Variant

In this problem we ask you to analyze another variant of the capacity scaling algorithm for max flow. To setup the notation, let $G = (V, E)$ be a graph, and for each edge from u to v, let $c(u, v)$ be the capacity and $f(u, v)$ be the flow.

Like the algorithm from class, we will have a current capacity scale D (which, in this algorithm, will always be a power of two). However, rather than modifying Ford-Fulkerson to use the D-residual graph of G, we will construct a new graph G_D by rounding the capacity of each (directed) edge down to the nearest multiple of D. We will then run Ford-Fulkerson unmodified–that is, using the standard notion of residual graph–on G_D, so that the inner iterations use the residual graph of G_D, rather than the D-residual graph of G! Specifically, the algorithm is as follows:

Algorithm 1 Scaling-Max-Flow-II

1: **let** D be the largest power of two less than or equal to U
2: $f \leftarrow 0$
3: **while** $D \geq 1$ **do**
4: Construct G_D ▷ capacity of each edge (u, v) is set to $D\lfloor \frac{c(u,v)}{D} \rfloor$
5: $f \leftarrow$ Ford-Fulkerson(G_D, f) ▷ run Ford-Fulkerson on G_D initialized with f
6: $D \leftarrow D/2$
7: **end while**

In Line 5, we assume that Ford-Fulkerson(G_D, f) begins with flow f and augments it until it reaches the max flow of G_D, then returns that flow vector. Note that replacing D with $D/2$ will only increase the capacities of edges in G_D, so f will still be a valid flow for the new G_D.

Prove that this algorithm terminates with a max flow in time $O(m^2(1 + \log U))$, just like the capacity scaling algorithm from class.