CSE 202: Design and Analysis of Algorithms Spring 2016

Problem Set 3

Instructor: Kamalika Chaudhuri **Due on:** Fri May 20, 2016

Instructions

- For your proofs, you may use any lower bound, algorithm or data structure from the text or in class, and their correctness and analysis, but please cite the result that you use.
- Problem 1 is worth 18 points (1 point for correctness, 2 for justification, per sub-question). Problem 2 is worth 12 points.
- If you do not prove that your algorithm is correct, we will assume that it is incorrect. If you do not provide an analysis of the running time, we will assume you do not know what the running time is.

Problem 1

The following statements may or may not be correct. For each statement, if it is correct, provide a short proof of correctness. If it is incorrect, provide a counter-example to show that it is incorrect.

1. In the capacity-scaling algorithm, an edge e can be present in the residual graph $G_f(D)$ in at most one D-scaling phase.
2. The max-flow problem with node capacities is defined as follows. We are given a directed graph $G = (V, E)$ with source s, sink t and capacities $c_v \geq 0$ for all nodes $v \in V$ (but no edge capacities c_e). A flow f is feasible in G if for all v other than s and t, the total flow into node v is at most c_v, the total flow into node v is equal to the total flow out of node v, and the size of a feasible flow f is the total flow out of s. Given this input, the max-flow problem with node capacities is to compute a feasible flow of maximum size.

   Computing a max-flow with node capacities can be reduced to the regular max-flow problem.
3. Let f and h be a preflow and a labeling compatible with the preflow. Let (u, v) be an edge in the original graph G, and let $h(v) = 4$ and $h(u) = 2$. Then, $f(u, v) = 0$.
4. Suppose you have already computed a maximum (s, t)-flow f in a flow network G with integer capacities. Let k be an arbitrary positive integer, and let e be an arbitrary edge in G whose capacity is at least k. Now suppose we increase the capacity of e by k units. Then, the maximum flow in the updated graph can be computed in $O(|E|k)$ time.
5. If all directed edges in a network have distinct capacities, then there is a unique maximum flow.
6. 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.

Problem 2

In a public building such as a movie theater, it is important to have a plan of exit in the event of a fire. We will design such an emergency exit plan in this question using max-flows. Suppose a movie theater 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.)

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

2. Show how the same idea can be used when people are initially in multiple locations and there are multiple exits.

3. Finally, suppose that it takes different (but integer) amounts of time to cross different corridors or stairways, and that for each such corridor or stairway e, you are also given an integer $t(e)$ which is the number of seconds required to cross e. Now show how to transform your algorithm in Part (1) to find a fastest way to get everyone out of the building.