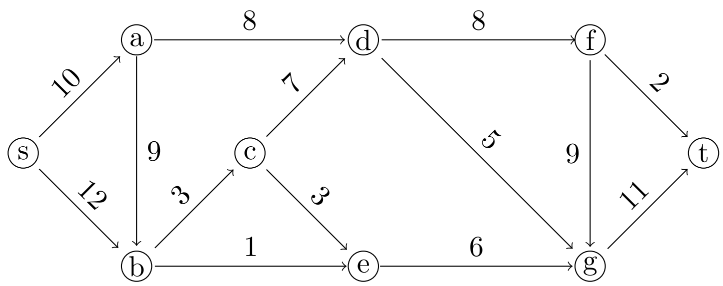
CSE421: Introduction to Algorithms	February 27, 2022
Homework 7	
Yin Tat Lee	Due: Mar 2, 2022 (before class)

Unless otherwise mentioned, you always need to show your algorithm's runtime and prove that it outputs the correct answer. See Homework Guideline on Ed for more details.

1. (10 Marks) Consider the following directed graph (edges e are labelled with capacities). Draw a maximum s-t flow for the graph below. Draw also the corresponding residual graph G_f for that flow. What is the minimum cut that corresponds to this maximum flow?



2. (10 Marks) Suppose you are given an instance of the max-flow problem with integer capacities c on a graph $G = (V, E)$. Every vertex v has an associated non-negative integer $d(v)$. Design an algorithm to find the maximum amount of flow that it's possible to send in to t, with the conservation constraint replaced by the constraint that the amount of flow in and out of a vertex v can differ by at most $d(v)$.

   Formally, we want to find a flow f that maximizes

   $$v(f) = \sum_{e \text{ in to } t} f(e)$$

   subject to the usual constraints of the max-flow problem, but instead of requiring for all v that $\sum_{e \text{ in to } v} f(e) = \sum_{e \text{ out of } v} f(e)$, we require

   $$\left| \sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) \right| \leq d(v)$$

   Your algorithm need only output $v(f_{\text{opt}})$, it's not necessary to output a flow. Let $M = \sum_{e \in E} c(e) + \sum_{v \in V} d(v)$. Your algorithm must run in time polynomial in M, m, and n.

3. (10 Marks) Give a polynomial time algorithm to find the minimum vertex cover in a bipartite graph.

   (a) Construct a flow network from the input bipartite graph just as in the maximum matching algorithm.

(b) Show that every min-cut in this flow network gives a vertex cover whose size is the same as the capacity of the cut.

(c) Show that every minimum sized vertex cover in the bipartite graph gives a cut whose capacity is the same as the size of the vertex cover.

(d) Write down the algorithm and prove that it works.

4. **(Extra Credit)** You are given a directed graph G with m edges and n vertices. For each edge e, we have an integer capacity u_e and an integer cost c_e. For any flow f, we define its cost be $\sum_e c_e f_e$. In general, there can be multiple s-t flows with maximum value. Show how to find a flow with minimum cost among these flows (s-t flows with maximum value).

   HINTS: First find any maxflow while ignoring the cost. Then lower its cost along some cycle containing s and t using the Bellman Ford Algorithm (Lecture 16).