Max Flow, Min Cut, and Matchings

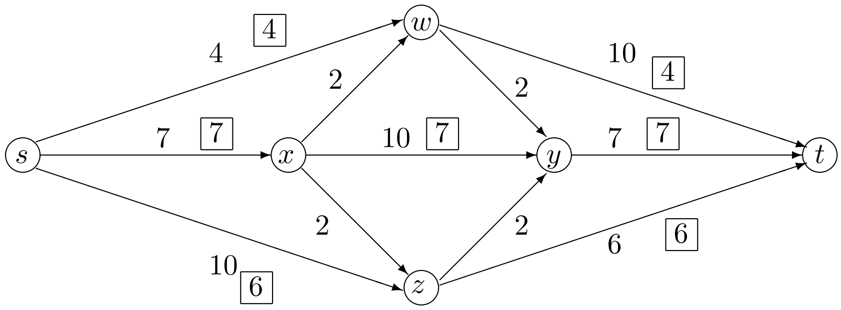
NB: Some problems below were taken from, or inspired by, exercises in the book *Algorithm Design* by Jon Kleinberg and Éva Tardos (Pearson International Edition, 2006).

1. The figure below shows a flow network on which an s-t flow is shown. The capacity of each edge appears as a label next to the edge, and the numbers in boxes give the amount of flow sent on each edge. (Edges without boxed numbers have no flow being sent on them.)

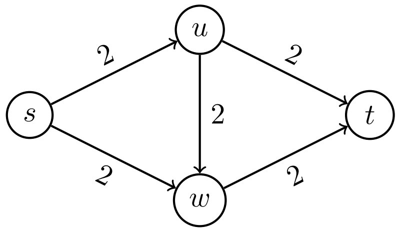
   (a) What is the value of this flow?

   (b) Is this a maximum s-t flow in this graph? If not, find a maximum s-t flow.

   (c) Find a minimum s-t cut. (Specify which vertices belong to the sets of the cut.)



2. Find *all* minimum s-t cuts in the following graph. The capacity of each edge appears as a label next to the edge.

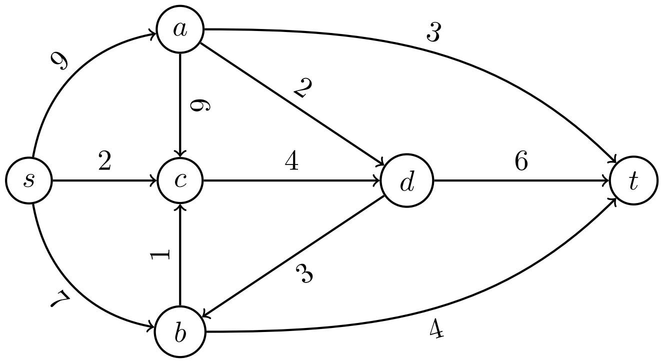


3. Consider the flow network H below with source s and sink t. The edge capacities are the numbers given near each edge.

(a) Find a maximum flow in this network.

Once you have done this, draw a copy of the original network H and clearly indicate the flow on each edge of H in your maximum flow.

(b) Find a minimum s-t cut in the network, i.e. name the two (non-empty) sets of vertices that define a minimum cut.



4. Network flows come up in dealing with natural disasters and other crises, since treatment of injured people and/or evacuation of large numbers of people without overloading hospitals and roads is necessary.

So consider the following scenario: Due to large-scale flooding in a region, paramedics have identified a set of n injured people distributed across the region who need to be rushed to hospitals. There are k hospitals in the region, and each of the n people needs to be brought to a hospital that is within a half-hour's drive to their current location. (So different patients will be able to be served by different hospitals depending upon the patients' locations.)

However, overloading one hospital with too many patients at the same time is undesirable, so we would like to distribute the patients as evenly as possible across all the hospitals. So the paramedics (or a centralised service advising the paramedics) would like to work out whether they can choose a hospital for each of the injured people in such a way that each hospital receives at most $\lceil \frac{n}{k} \rceil$ patients.

Describe a procedure that takes the given information about the patients' locations (hence specifying which hospital each patient could go to) and determines whether a balanced allocation of patients is possible (i.e. each hospital receives at most $\lceil \frac{n}{k} \rceil$ patients).

What is the asymptotic running time of your procedure (in terms of n and k)?

5. Suppose you live with $n-1$ other people in an off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the entire group exactly once, so that someone different cooks on each night.

   Due to scheduling constraints (concerts, sports, etc), each person is unable to cook on certain nights, so deciding on who is cooking on each night appears to be a tricky task. Suppose we label the people in the flat $\{p_1, p_2, \ldots, p_n\}$ and the nights $\{d_1, d_2, \ldots, d_n\}$.

   Then for each person p_i, there is a set of nights $S_i \subseteq \{d_1, d_2, \ldots, d_n\}$ where p_i is unable to cook.

   A *feasible dinner schedule* is an assignment of each person in the flat to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j then $d_j \notin S_j$.

   Describe an algorithm to determine if there is a feasible dinner schedule or not.

   What is the running time of your procedure?

6. Suppose you are given a flow network with unit capacity edges, i.e. you have a directed graph $G = (V, E)$, a source vertex s and a sink vertex t, and each edge has capacity 1. You are also given an integer parameter k.

   Your goal is to delete a set of k edges from G in order to reduce the maximum s-t flow as much as possible. In other words, you want to find a subset of edges $F \subseteq E$ consisting of *exactly* k edges to delete from G, giving a new flow network G', and the maximum s-t flow in G' is as small as possible.

   Describe an efficient algorithm for performing this task.