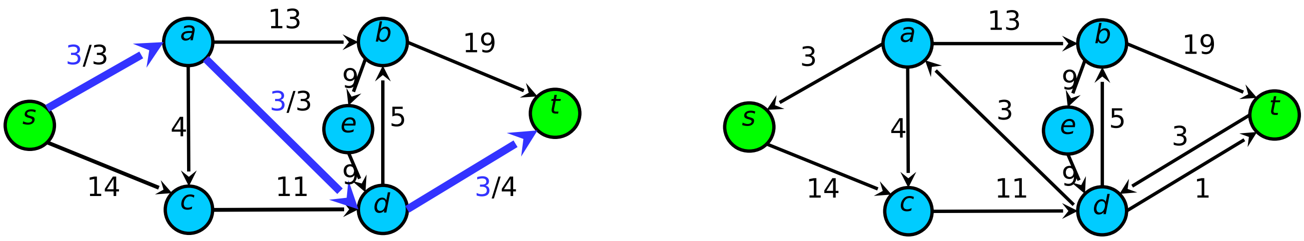
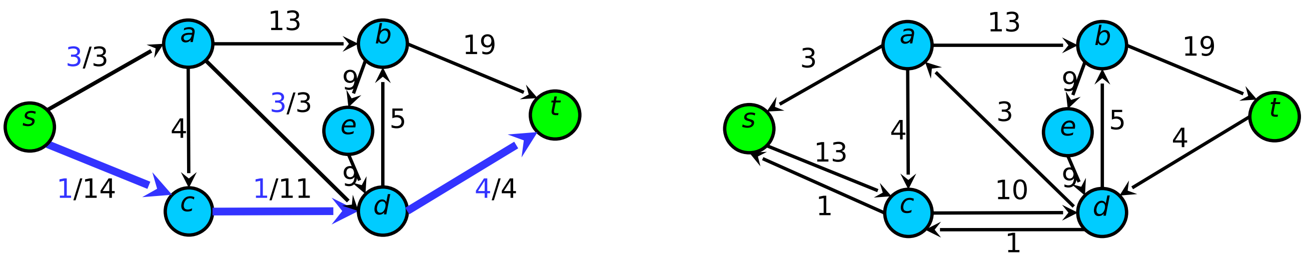
Lecture 3 exercise solutions

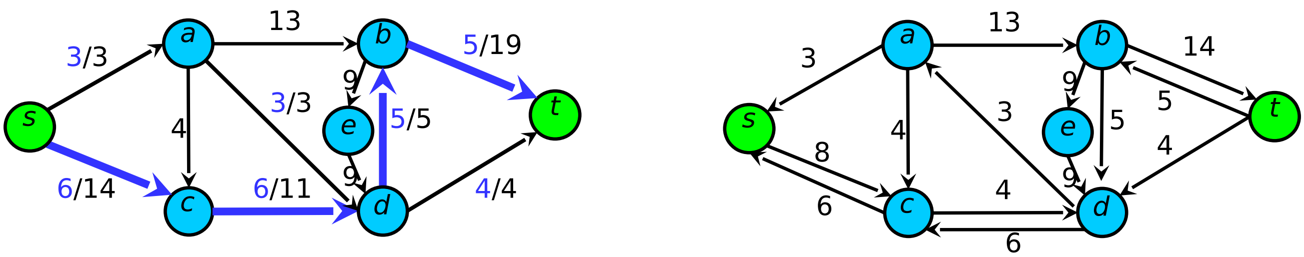
1.



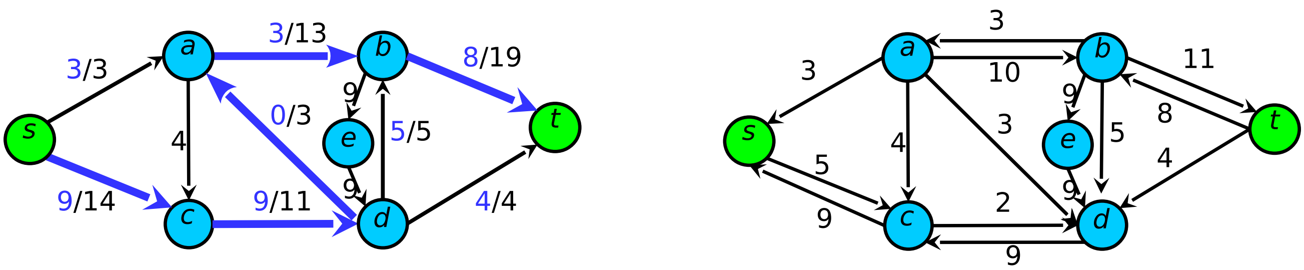
Augmentation 1 (path *sadt*, flow sent 3):



Augmentation 2 (path *scdt*, flow sent 1):



Augmentation 3 (path *scdbt*, flow sent 5):



Augmentation 4 (path *scdabt*, flow sent 3). Note, how the flow is canceled on (a, d):

Maximal flow found: 12

CLRS4 24.2-4

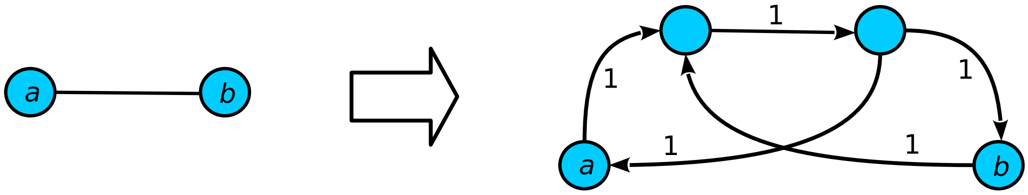
$(\{s, v_1, v_2, v_4\}, \{v_3, t\})$ is the min-cut corresponding to the max-flow. Third augmenting path, in (c), cancels flow on (v_2, v_1) and (v_3, v_2).

CLRS4 24.1-7.

In G', each vertex v from V is transformed into a pair of vertices and an edge connecting them (v_s, v_e) with a capacity $l(v)$. All incoming edges of v connect to v_s and all outgoing edges of v connect to v_e. The transformed graph has $|V'| = 2|V|$ and $|E'| = |E| + |V|$.

CLRS4 24.1-6.

First, we start by modeling the street map as an undirected graph with vertices as intersections and edges as streets (blocks). Just transforming a street into one directed edge with a capacity of 1 obviously does not work as we want to be able to go both ways on the street. Modeling a street as two anti-parallel edges with capacities of 1 also does not work (but see below!) as we want to be able to use the street only once (in one of the two directions). Here is instead how we want to transform a street:



Just one unit of flow can be sent from *a* to *b* or from *b* to *a*. Home and school intersections are respectively source and sink vertices. If the value of a maximum flow in the network is smaller than 2, then it is not possible to send both children to school.

Less intuitively, one *can* actually use two anti-parallel edges with capacities of 1 to model a two-way street! Any flow that uses both edges between some pair of vertices can be modified by just *removing* the flow on these two edges. This 1) leaves the flow valid, as the flow conservation is not violated for both of the vertices (minus one unit of inflow and minus one unit of outflow) and 2) the value of the flow in the network is not affected: even if one of the vertices is the source or the sink, this removal of flow simultaneously increases the flow value by one and decreases it by one. In this way we can remove all instances of flow on anti-parallel edges without changing the value of the flow. Thus, the maximum flow of two or more in such a network indicates the possibility of sending both children to school.

CLRS4 29.2-5.

Here is a *binary linear program* for maximum matching in a bipartite graph $G=(L \cup R, E)$

$$
\begin{aligned}
&\text{maximize} \quad \sum_{(i,j)\in E} x_{ij} \\
&\text{subject to} \\
&\qquad \sum_{j\in R} x_{ij} \leqslant 1, \forall i \in L \\
&\qquad \sum_{i\in L} x_{ij} \leqslant 1, \forall j \in R \\
&\qquad x_{ij} \in \{0,1\} \text{ for } i \in L \text{ and } j \in R
\end{aligned}
$$

Although beyond the scope of this course, we can observe that this is an integer linear program and thus, in general, an NP-hard problem. Nevertheless, the integrality requirement (last line) can be dropped, replacing it wih $0 \leqslant x_{ij} \leqslant 1,$ for $i \in L$ and $j \in R$. This, so-called *relaxation*, gives a standard-form linear program that can be solved with polynomial-time algorithms. If some of the returned x_{ij} are not integral, they can be adjusted to be integral (in polynomial time). Of course, all that can be avoided and efficient polynomial-time algorithms described in sections CLRS4 24.3 or 25.1 can be used instead.