CSC373 Network Flow Examples

Note: These questions are all from chapter 26 of CLRS 3rd edition.

Table of Contents:

Question 26.1-1:

There are 2 main scenarios to consider:

1. Paths that don't go through (u,v) in G.
2. Paths that go through (u,v) in G.

For the first case, the flow remains the same for all paths that don't go through (u,v) in G.

For the second case, if (u,v) is the bottleneck for a path that went through it in G, then (u,x) and (x,v) will be the bottleneck and because $c(u,v) = c(u,x) = c(x,v)$, the bottleneck value doesn't change. Hence, the flow doesn't change.

Now, if (u,v) isn't the bottleneck, then the flow doesn't change as $c(u,v) = c(u,x) = c(x,v)$. The bottleneck value doesn't change.

Hence, in all scenarios, the flow remains the same.

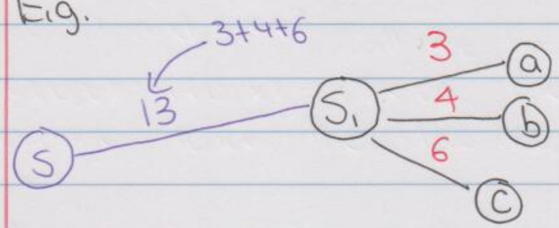
Question 26.1-2:

We can connect S, the supersource, with all source nodes and t, the supersink, with all sink nodes.

Let $S_1, S_2, \ldots, S_n$ be all source nodes.
Let $T_1, T_2, \ldots, T_m$ be all sink nodes.

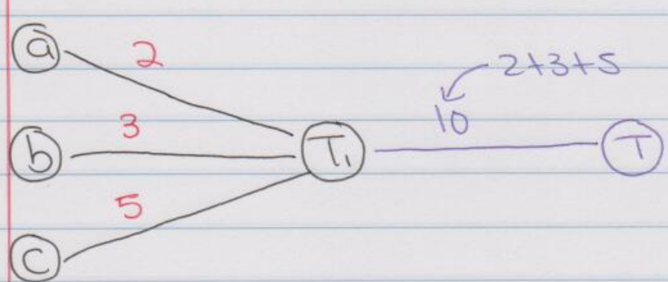
For each (S, S_i) edge, let the capacity be equal to the sum of the capacities that leave S_i.

E.g. I.e. $cap(S, S_i) = \sum_{v \in V} cap(S_i, v)$



For each (T_i, T) edge, let the capacity be equal to the sum of the capacities that enter T_i.

E.g.



Note: You can also make each $cap(S, S_i) = \infty$ and $Cap(T_i, T) = \infty$.

Question 26.1-3

If there's no $s \leadsto u \leadsto t$ path, then at least 1 of the following must be true:

1. There's no path from s to u.
2. There's no path from u to T.

Consider case 1. If there's no path from s to u, then there's no flow coming in since S is where the flow comes from.

Consider case 2. If there's no path from u to T, then there's no place for flow to discharge since T is the sink. Hence, the flow must be 0.

Question 26.1-6:

The question, in effect, is asking us to show how to use max-flow to solve edge disjoint path problem.

Let S be the home.
Let T be the school.
Let $cap(u,v) = 1 \ \forall (u,v) \in E$.
I.e. Set the capacity of each edge to be 1.
We need to do this to prevent the same edge from being used multiple times.

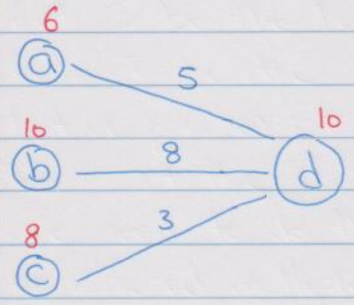
If there's a max flow value of k, then we know there are k disjoint paths.

Hence, if we can get a max flow value of at least 2 then we know there are at least 2 edge disjoint paths.

Question 26.1-7:

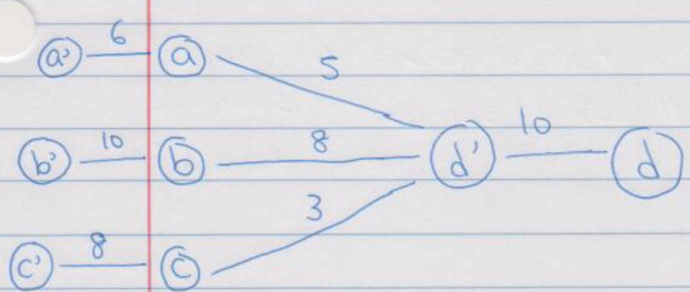
For each vertex/node v, create an intermediary vertex/node v' s.t. $cap(v', v) = \ell(v)$.

E.g. Suppose we have the following graph in G:



Blue nums indicate edge capacities.
Red nums indicate vertex capacities.

Now, we'll have this graph:



In this new system, we have $2|V|$ vertices and $|V| + |E|$ edges.

Question 26.2-6

Let S be the source node and T be the sink node.
Let $cap(S, S_i) = P_i$
Let $cap(T_j, T) = Q_j$

Question 26.2-8

The F-F algo stops when there are no more S-T paths.
There are 3 cases to consider:

1. $cap(s, a)$ $cap(u,v)$ $\forall (u,v)$ for each edge in this S-T path.
E.g.

$$S \xrightarrow{10} A \xrightarrow{8} T$$

Then, when we augment the graph/path, the bottleneck edge will point towards S, removing the S-T path.

2. $cap(s, a) < cap(u,v)$ $\forall (u,v)$ for each edge in this S-T path
E.g.

$$S \xrightarrow{2} A \xrightarrow{5} T$$

When we augment the path, we won't have a (A,S) edge but bc the (S,A) edge is used up, we can think that it disappeared.

3. $cap(s,a)$ is somewhere in the middle.
Then, the bottleneck edge will point to s after that path is augmented.

$\therefore$ F-F still generates the max flow.