CSE 551 Homework 5 Solutions

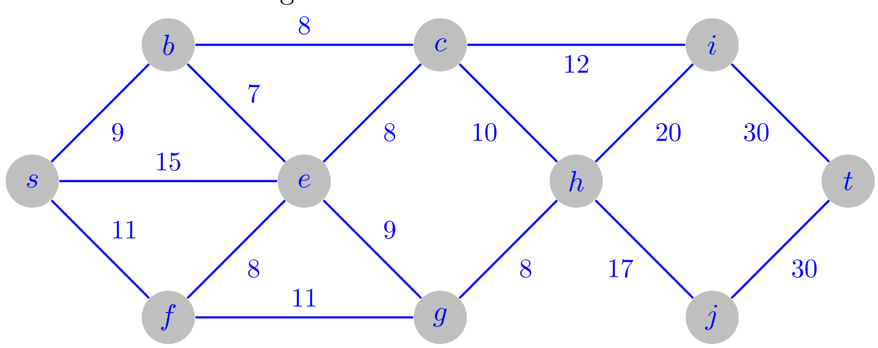
October 2021

1 Question 1

Run the Ford-Fulkerson algorithm on Figure 1 to find the max s-t flow.

1. Use any augmenting path you want.
2. Use capacity scaling.

Figure 1: Problem 1 Network

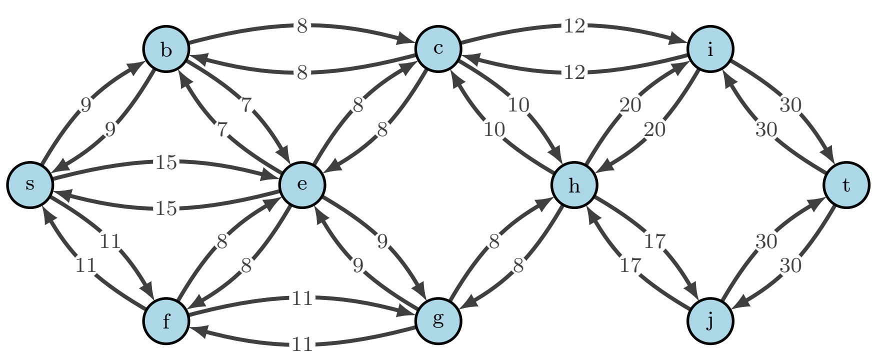


Solution:

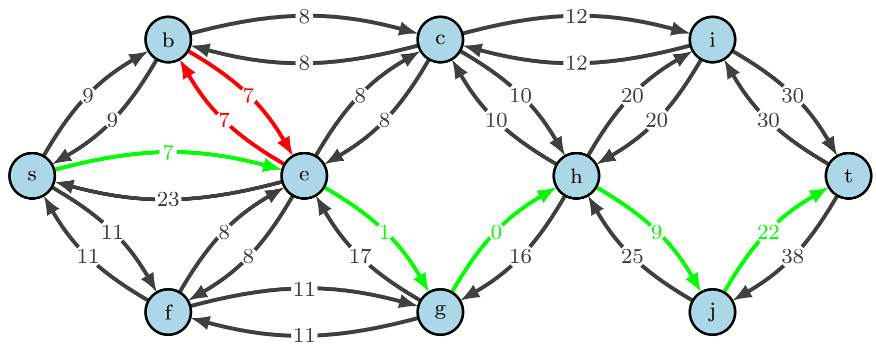
1. There really is not too much of a difference between the generic Ford-Fulkerson method and using the capacity scaling method, there are plenty of options for paths to augment even when using scaling capacity as seen in the 1.2.
2. The largest value in this graph is 30, so the scaling factor δ is the largest power of 2 that is strictly less than 30, i.e., 16.

   However, notice that every edge leaving out of s has capacity less than 16, so in fact you will need to halve the scaling factor to 8 before anything can be done. Below is *one* way of solving the problem.

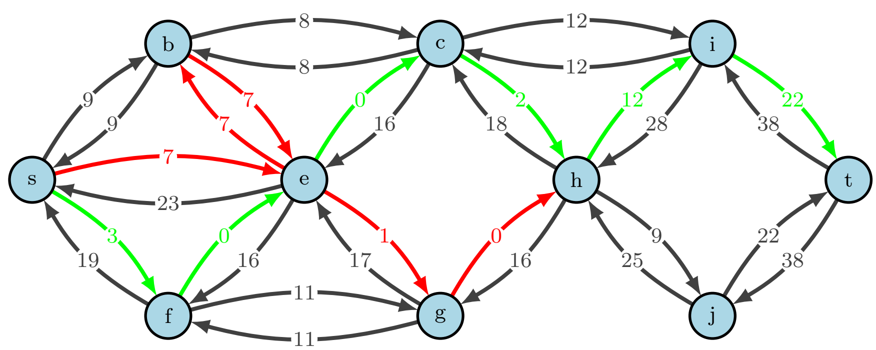
   Create the residual graph:



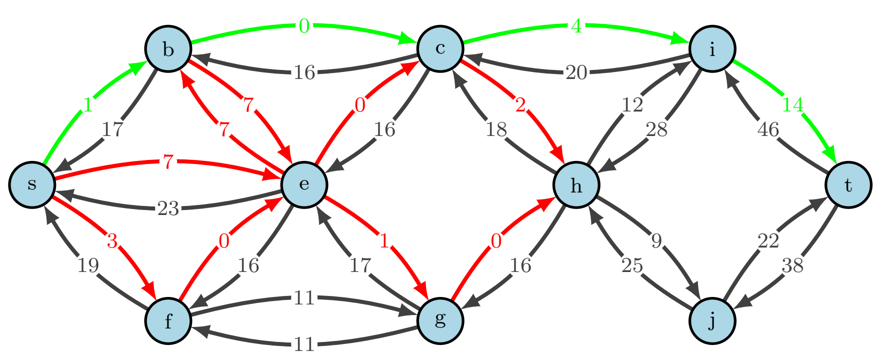
Take the augmenting path in green (also, since the scaling factor is 8 forget the edges with values of 7:



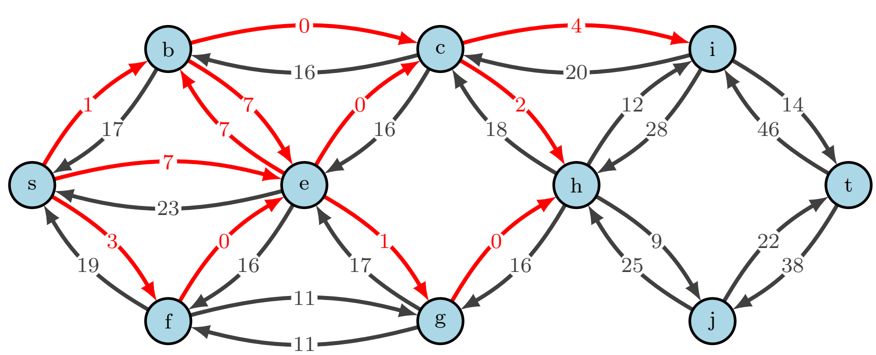
Take the folloiwing augmenting path in green (remember you cannot follow s to e as 7 is less than hte current scaling factor!)



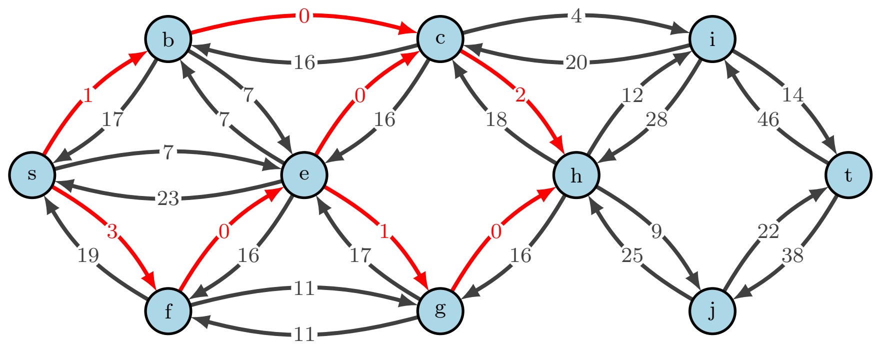
Augment the path in green.



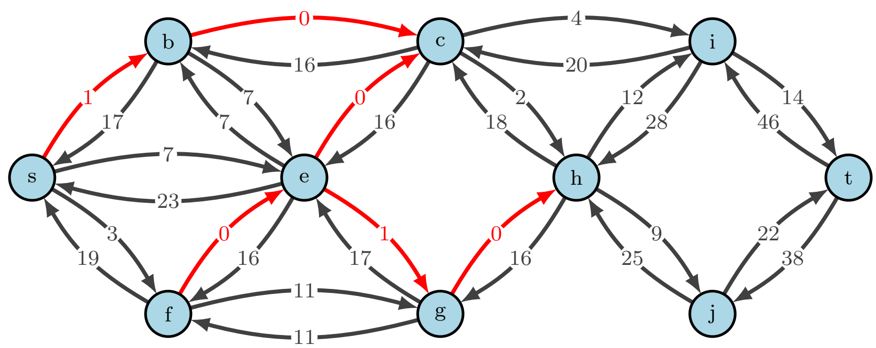
No augmenting paths.



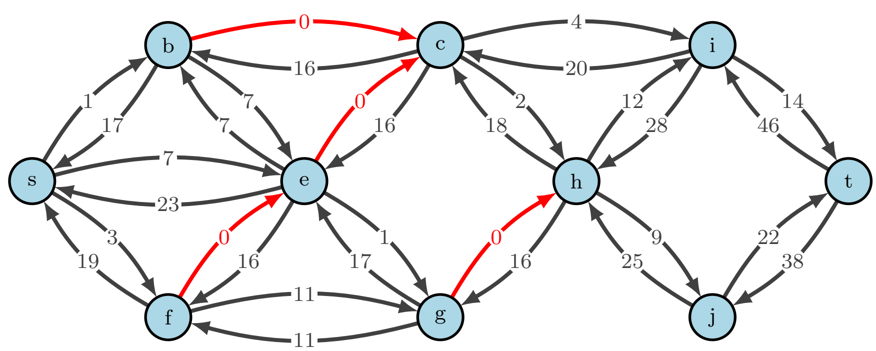
Halve scaling factor ($\Delta = 4$), still no augmentable paths!



Halve scaling factor ($\Delta = 2$), still no augmentable paths.



Halve scaling factor ($\Delta = 1$), still no augmentable paths, so we are done.



The sum of the flows going out of s is $8 + 8 + 8 = 24$.

2 Question 2

1. Let (G, s, t, c), $c : E \to \mathbb{N}^+$ be a flow network. If f is a maximum s-t flow in G, is it true that f saturates every edge out of s with flow (i.e., for all edges e out of s, we have $f(e) = c(e)$). Justify your answer.

2. Let (G, s, t, c), $c : E \to \mathbb{N}^+$ be a flow network. Let (A, B) be a minimum s-t cut with respect to these capacities $c(e) : e \in E$. Consider the flow network (G, s, t, c') where $c'(e) = c(e) + 1$. Is (A, B) still a minimum s-t cut with respect to c'? Justify your answer.

Solution:

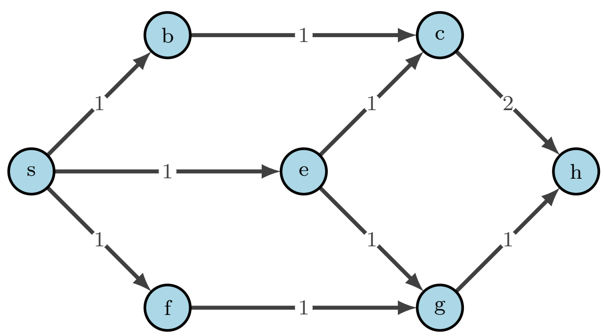
1. This is not true, consider the following network (G, s, t, c):

$$\begin{aligned} V(G) &= \{s, a, t\} \\ E(G) &= \{(s, a), (a, t)\} \\ c((s, a)) &= 10 \\ c((a, t)) &= 1 \end{aligned}$$

   The max flow from s to t is 1 because the edge (a, t) can hold at most one unit of flow going through it. However, this also means that the flow going through (s, a) is only 1 and not fully saturated at 10 units of flow.

2. This is also not true. Consider the following graph described in Figure 2. The set $\{s\}$ is a minimum cut, however if you transform the graph, as per the question, it is not longer a minimum cut. as $V(G) - \{t\}$ is a cut with value 5 and $\{s\}$ has a capacity of 6 in the updated graph.

Figure 2: Problem 2.2 Counterexample



3 Question 3

Because of COVID-19 many people have phoned in saying they are sick. Doctors have identified that n people are high risk and very likely have COVID-19. We want to try our best not to overfill the hospitals while at the same time we only want people have to drive at most 30 minutes to a hospital. Suppose there are k hospitals in the region with capacities $c_1, c_2 \ldots, c_k$ and people have indicated which hospitals are 30 minutes away from them. Devise a polynomial algorithm that determines if people can be distributed to each hospital in a *balanced* manner, i.e. is it possible to distribute people so that no more than $\lceil n/k \rceil$ people are in each hospital?

Solution:

First we construct a graph G that "maps" people to hospitals they are eligble to go to. Denote the n people as p_i such that $i = 1 \ldots n$. Denote the k hospitals as h_j such that $j = 1 \ldots k$. Create the edge (p_i, h_j) if and only if p_i can reach hospital h_j in 30 minutes.

Now, we proceed to add on a source and sink onto G. For every (p_i, h_j) constructed above, define $c((p_i, h_j)) = 1$. Add a node s and a node t. Create the edges (s, p_i) for $i = 1 \ldots n$. And set $c((s, p_i)) = 1$. Lastly add edges (h_j, t) for $j = 1 \ldots k$ with $c((h_j, t)) = \min\{\lceil n/k \rceil, c_j\}$.

Theorem 3.1. *Running the Ford-Fulkerson Algorithm will distribute the people correctly if possible. If it is not possible, then the max flow will be $< n$.*

Proof. Consider a person represented by node p_i. s can supply one unit of flow through the node, which then may be distributed through a hospital node that the person can reach. So, if every person can distribute their flow through the hospital, because each edge capacity is integral, the integrality constraints theorem implies p_i must either give all 1 unit to a particular hospital and none to the others. This corresponds to them physically being able to attend the hospital. What remains to show is that hospital satisfy the equidistribution criterion. So, suppose otherwise, that a hospital is not equidistributed at the end. If so, some hosptal h_j must channel a unit flow from more than $\lceil n/k \rceil$ units of flow from people, but this is impossible as that would imply $c((h_j, t)) > \lceil n/k \rceil$, but by construction $c((h_j, t)) \leq \lceil n/k \rceil$. $\square$

First, observe the construction of G does not require more than polynomial more edges and vertices than the size of the input. Second, Ford-Fulkerson algorithm can be ran in polynomial time. Therefore, we have devised a polynomial time algorithm for the problem.

4 Question 4

Given a network (G, s, t), give a polynomial time algorithm to determine if G has a unique minimum cut.

Solution: WLOG assume the network has integral flows. Compute a minimum $s - t$ cut (A, B) ($s \in A$, $t \in B$) using the Ford-Fulkerson algorithm. Denote the capacity of the minimum cut as C. Call the set of edges (x, y) (with $x \in A$ and $y \in B$) O. For each edge $(x, y) \in O$, modify the capacity of (x, y), $c_{(x,y)}$ to be $c_{(x,y)} + \frac{1}{|O|+10}$.[1]

Theorem 4.1. *If and only if after running the Ford-Fulkerson algorithm on the modified graph, there is a flow f, such that $|f| < C + \frac{|O|}{|O|+10}$, the number of paths in the original graph was not unique.*

Proof. ($\Rightarrow$) Suppose that after running the Ford-Fulkerson algorithm there is a flow f with $|f| < C + \frac{|O|}{|O|+10}$. But as all edges capacities in the original graph were integral and the sum of the capacities of edges from A to B is less than 1, it must be the original flow had value $|f| = C$, and therefore is minimal.
($\Leftarrow$) Suppose there was another minimal cut (A', B') that was not detected. So it had value C before the weights were updated. At worst this cut had all but one edge O, which left A' and entered B'. But then under the new capacities, the new graph would have a capacity of $C + \frac{|O|-1}{|O|+10} < C + \frac{|O|}{|O|+10}$. $\square$

As Ford-Fulkerson is only ran a constant number times on networks that are on the order of the input, the process finishes in polynomial time.

5 Question 5

Show that the Ford-Fulkerson algorithm may not terminate when irrational edge capacities are allowed.

(Hint: Consider Figure 3 where all edge capacities have $\frac{1}{1-\sigma}$ where $\sigma = \frac{\sqrt{5}-1}{2}$. Except let $c((x_1, y_1)) = 1$, $c((x_2, y_2)) = \sigma$, and $c((x_4, y_4)) = \sigma^2$. It might be useful to prove that $\sigma^n = \sigma^{n+1} + \sigma^{n+2}$).

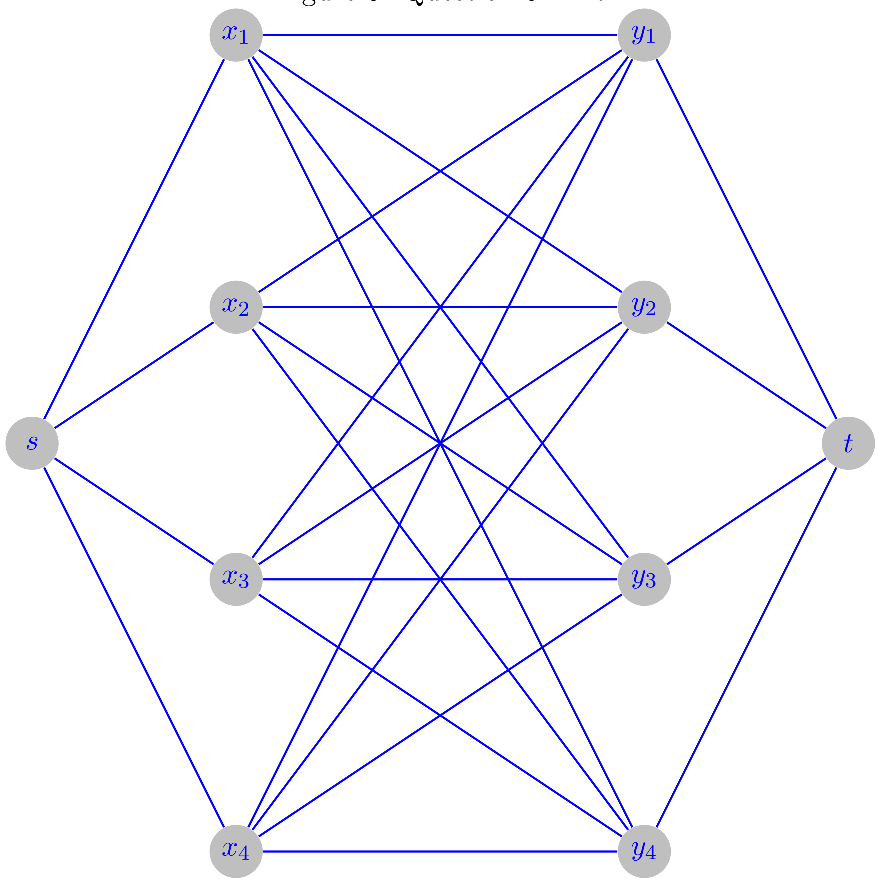
Solution:

First let us prove the hint:

Lemma 5.1. $\sigma^n = \sigma^{n+1} + \sigma^{n+2}$.

[1]There are other valid choices for what to add to $c_{(x,y)}$, but you must ensure that the result is rational (lest it may not terminate), and the updated sum is less than 1

Figure 3: Question 5 Hint



Proof. We prove this by induction. For the base case, we claim that

$$\sigma^0 = \sigma^1 + \sigma^2$$

or, in other words, $1 = \sigma^1 + \sigma^2$. So,

$$\begin{aligned}
\sigma + \sigma^2 &= \frac{\sqrt{5}-1}{2} + \left(\frac{\sqrt{5}-1}{2}\right)^2 \\
&= \frac{\sqrt{5}-1}{2} + \frac{(\sqrt{5}-1)^2}{4} \\
&= \frac{2\sqrt{5}-2}{4} + \frac{5-2\sqrt{5}+2}{4} \\
&= \frac{2\sqrt{5}-2+5-2\sqrt{5}+2}{4} \\
&= \frac{4}{4} = 1
\end{aligned}$$

Now consider $\sigma^{n+1} = \sigma(\sigma^n)$. By the inductive hypothesis $\sigma^n = \sigma^{n+1} + \sigma^{n+2}$, therefore

$$\sigma(\sigma^n) = \sigma(\sigma^{n+1} + \sigma^{n+2}) = \sigma^{n+2} + \sigma^{n+3}$$

□

Now observe the problem requires the Ford-Fulkerson algorithm and *does not* require any specific version of the Ford-Fulkerson. Thus, if there are multiple choices for paths that can be augmented, one is allowed to freely choose between the paths.

The goal is to indefinitely generate new terms of the sequence $\sigma^{n+2} = \sigma^n - \sigma^{n+1}$. Let's first set up some terminology. Recall that all edges have capacities $\frac{1}{1-\sigma}$ except for $c((x_i, y_i))$, $i = 1 \ldots 4$. Call these edges *special.* In order to generate terms of the sequence, we claim that for any n we can always have the residual flow of the special edges to be some combination of either 0, σ^{n+1}, σ^{n+2}, or σ^{n+2} as long as the augmenting paths is chosen carefully enough.

This proof also uses induction. For the base case ($n = 0$), we must show that residual capacity of (x_i, y_i), $i = 1 \ldots 4$ is some combination of 0, σ^1, σ^2, σ^2. Without having chosen any augmenting paths we have:

$$
\begin{aligned}
c((x_1, y_1)) &= 1 \\
c((x_2, y_2)) &= \sigma^1 \\
c((x_3, y_3)) &= \sigma^2 \\
c((x_4, y_4)) &= \sigma^2
\end{aligned}
$$

We are missing 0. To rectify this, augmenting along the path sx_1y_1t gives 0 remaining capacity in (x_1, y_1).

By induction we assume there is some permutation ρ such that the residual capacity of $(x_{\rho(i)}, y_{\rho(i)})$, is some combination of 0, $\sigma^n, \sigma^{n+1}, \sigma^{n+1}$. WLOG impose the remaining capacity of $(x_{\rho(1)}, y_{\rho(1)})$ is 0, $(x_{\rho(2)}, y_{\rho(2)})$ is σ^n, $(x_{\rho(3)}, y_{\rho(3)})$ is σ^{n+1}, and $(x_{\rho(4)}, y_{\rho(4)})$ is σ^{n+1}. Supply flow through $sx_{\rho(2)}y_{\rho(2)}x_{\rho(3)}y_{\rho(3)}t$ This gives the "special" arcs the following residual capacities:

$$
\begin{aligned}
&(x_{\rho(1)}, y_{\rho(1)}) \to 0 \\
&(x_{\rho(2)}, y_{\rho(2)}) \to \sigma^n - \sigma^{n-1} = \sigma^{n+2} \\
&(x_{\rho(3)}, y_{\rho(3)}) \to 0 \\
&(x_{\rho(4)}, y_{\rho(4)}) \to \sigma^{n+1}
\end{aligned}
$$

Still not quite right. Instead send flow along $sx_{\rho(2)}y_{\rho(2)}y_{\rho(1)}x_{\rho(1)}y_{\rho(3)}x_{\rho(3)}y_{\rho(4)}t$. Going through $(x_{\rho(2)}, y_{\rho(2)})$ is going to supply σ^{n+2} units of flow and bring $(x_{\rho(2)}, y_{\rho(2)})$ down to 0 units of flow. Then going through $(y_{\rho(1)}, x_{\rho(1)})$ will now have σ^{n+2}, thus, the two edges swap residual capacities. Then flowing from $(y_{\rho(3)}, x_{\rho(3)})$ creates another σ^{n+2} flow. Last must distribute the flow somehow to t somehow, which is done by following $x_{\rho(3)}y_{\rho(4)}t$.

One last thing to observe is that no edge that is *not* special does not fill up and cause any issues. The first path augmented ($sx_{\rho(2)}y_{\rho(2)}x_{\rho(3)}y_{\rho(3)}t$) adds σ^{n+1} in total. The second path ($sx_{\rho(2)}y_{\rho(2)}y_{\rho(1)}x_{\rho(1)}y_{\rho(3)}x_{\rho(3)}y_{\rho(4)}t$) add σ^{n+2} units of flow so, by Lemma 5.1, the network has increased by σ^n. So, at worse, the network should be able to support:

$$
\sum_{i=1}^{\infty} \sigma^n
$$

But this is a geometric series with value $\frac{1}{1-\sigma}$, the capacity of the non-special edges.