FIT3155: Week 12 Tutorial - Answer Sheet

(Scribe: Dinithi Sumanaweera)

Question 1

Maximize **z = x + 2y** using tableau method, subject to following constraints

$$\begin{aligned} 2x + 4y &\leq 12 \\ 4x + 3y &\leq 16 \\ x &\geq 0 \\ y &\geq 0 \end{aligned}$$

		1	2	0	0		
		x	y	s	t	R.H.S	θ
0	s	2	4	1	0	12	$\boxed{3}$
0	t	4	3	0	1	16	$16/3$
$c_j - z_j$		1	$\boxed{2}$	0	0	$Z=0$	
2	y	$1/2$	1	$1/4$	0	3	
0	t	$5/2$	0	$-3/4$	1	7	
$c_j - z_j$		0	0	$-1/2$	0	$Z=6$	

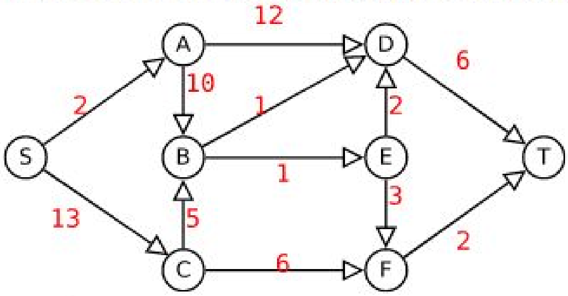
s leaves the basis

y enters the basis

Notice that x is still a non basic variable (i.e. x=0), and its coefficient in the objective function is 0 (i.e. cj-zj == 0). At this stage, the objective function value is 6 (i.e. Z=6), and no matter how much we increase x, it does not further increase Z. In that case, this problem has infinitely many solutions.

Question 2

Consider the following network with the edge capacities marked in red:

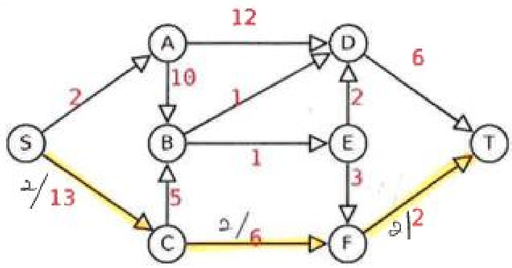


Find the maximum flow in this network. Also show the minimum cut.

Iteration 1

flow(e) = 0

G = residual graph.

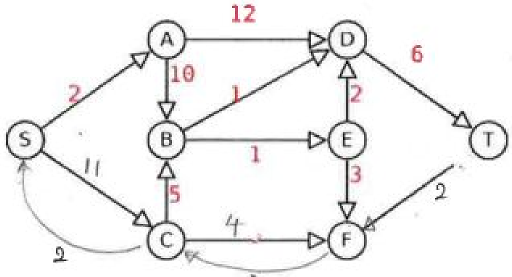


Augmented graph

Select path

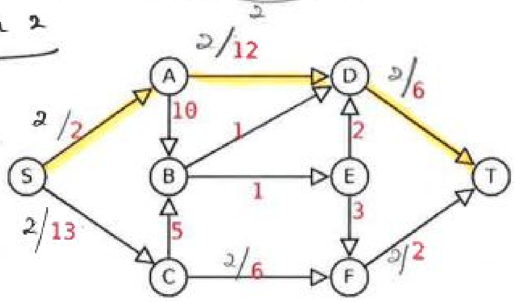
flow bottleneck

= min {13, 6, 2} = 2



Residual graph

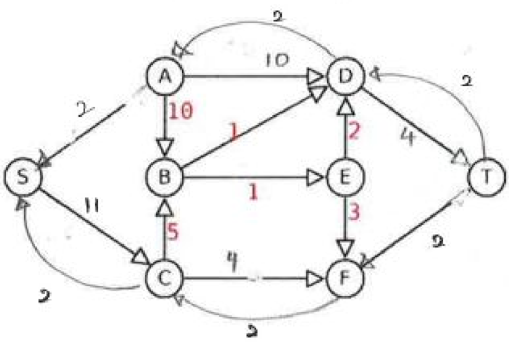
Iteration 2



Augmented graph

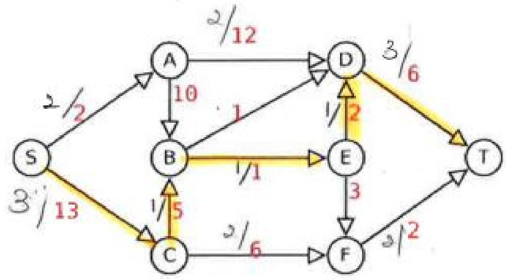
flow bottleneck

= min {2, 12, 6} = 2

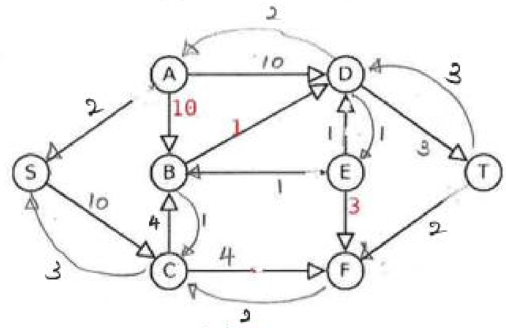


Residual graph

Iteration 3

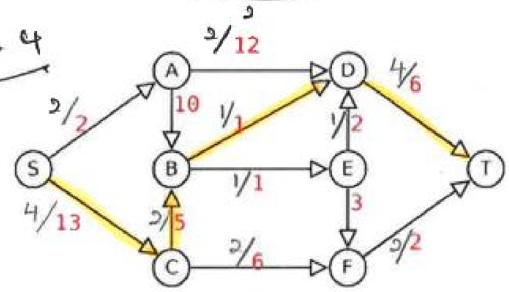


flow bottleneck
= min{13,5,1,2} = 1

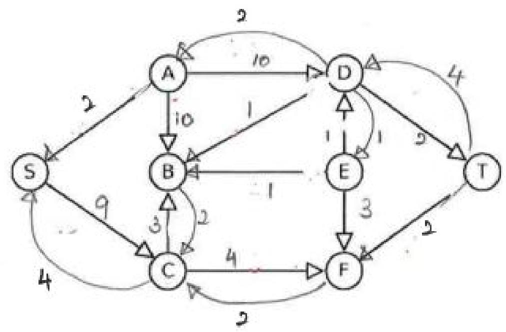


Residual network

Iteration 4



flow bottleneck
= min{13,5,1,6} = 1



Final
Residual network

Spannin

Max flow = 4+2 = 6

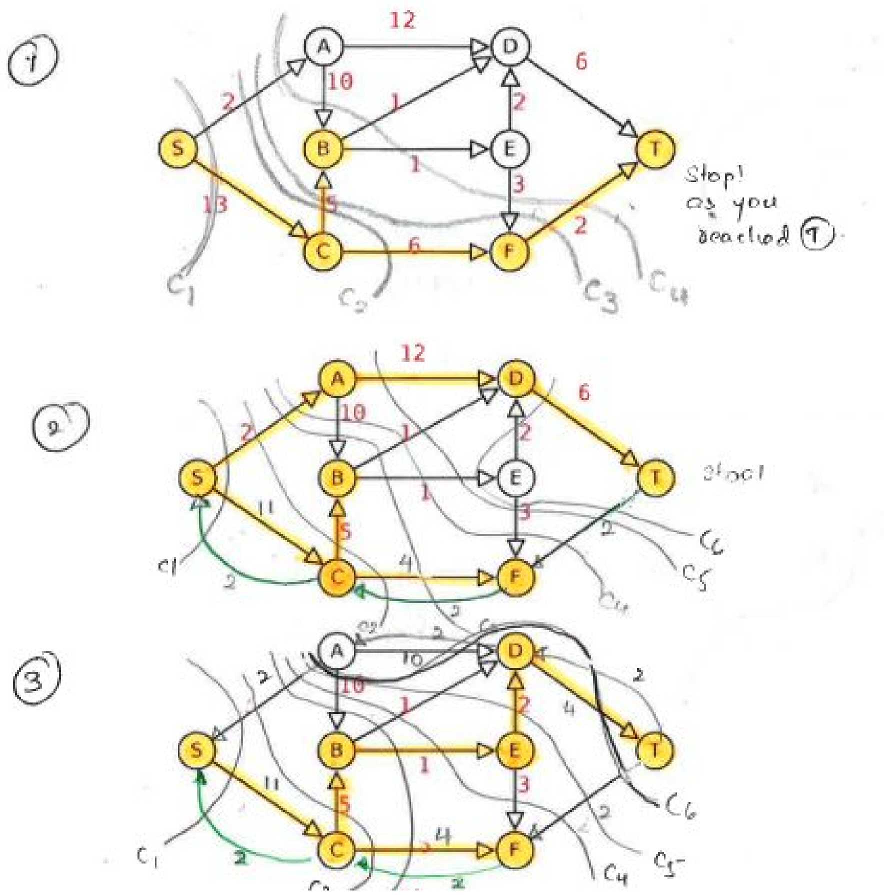
The path selection is done with **Edmonds-Karp (largest bottleneck)** method

Additional Note:

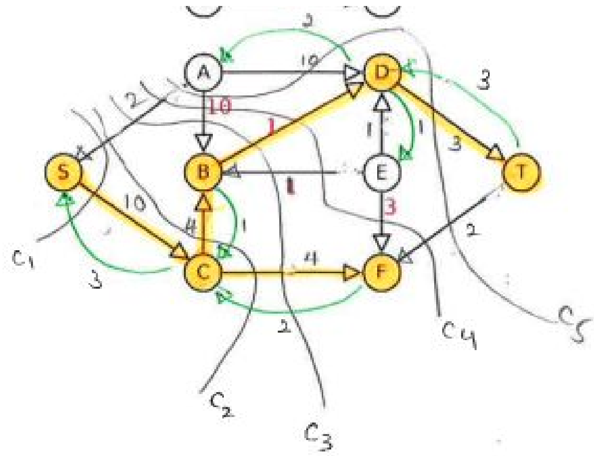
When implementing the Edmonds-Karp (largest bottleneck) augmentation for Ford-Fulkerson algorithm, to decide on the path for the augmented network, we first start growing a spanning tree over the residual network.

```
Consider the residual graph
Initialize set_1 = {source_node} and set_2 = {all other nodes}
while(sink_node is not reached){
   -  Make a cut crossing all edges of set_1 and consider all
      outgoing edges from that.
   -  Pick the edge with maximum capacity
   -  Move the corresponding node at the end of that edge which is
      currently in set_2, to set_1.
}
```

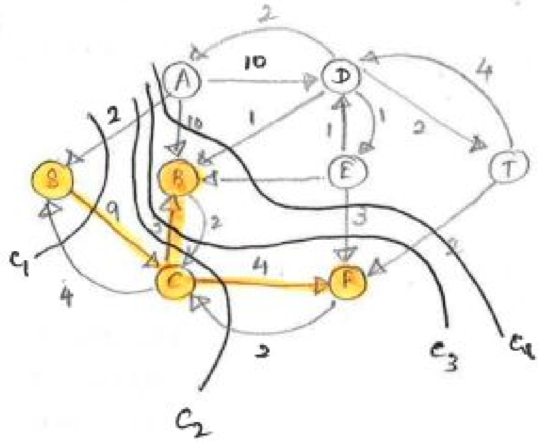
Once you reach the sink node, stop growing the spanning tree. Now you can choose the path from source to sink in this spanning tree as the path for the current iteration of Ford-Fulkerson algorithm, where you need to first get the flow bottleneck over that path and build the next augmented network and the corresponding residual network.



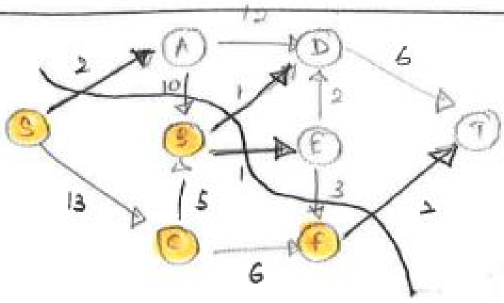
4



5



Can't proceed further. Stop!
(final)



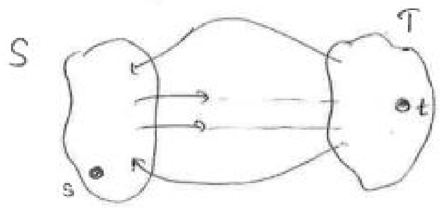
Min cut!
$2+1+1+2 = \underline{\underline{6}}$

Min cut capacity == max flow == 6

Note: When you stop at iteration 5 due to not being able to reach the sink anymore, it is because there are no more outgoing edges that cross the cut. In terms of the augmented flow network, this means all outgoing edges have been saturated (i.e. each edge accommodate their max possible flow which is equal to its capacity). This means you have reached the max flow in the network. Since max flow == min cut capacity, the corresponding cut is the min cut.

Question 3

Revise the theorem that proves that Min-cut in a flow network is equal to its Max-flow



Cut is represented by 2 sets.

s = source
t = sink

Net flow of the cut $(S,T) = Flow^{out}(S) - flow^{in}(S)$
/ Total flow

Capacity of a cut = How much I can push out of the cut / (S)

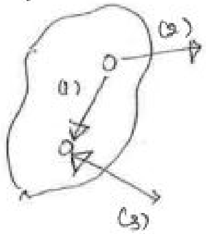
Considers only <u>outgoing</u> edges.

① Net flow of a cut == flow value of the network == flow(G)

$$flow(G) = Flow^{out}(s)$$

$$= \sum_{\forall x \in S} \left[Flow^{out}(x) - flow^{in}(x) \right] = flow^{out}(S) - flow^{in}(S)$$

3 types of possible edges.



For (1), one's $flow^{out}(x_1)$ will be a $flow^{in}(x_2)$ of another.

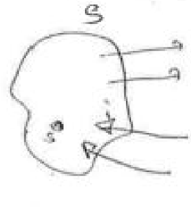
$\Rightarrow$ They cancel out.

Remaining will be (2) outgoing
(3) incoming edges to the set.

(2) Any flow of network

$$flow(G) \leq \text{Capacity of any cut}$$

$\Rightarrow$ $\begin{pmatrix}\text{min cut}\\ \text{Capacity}\end{pmatrix} = min[\text{Capacity (Cut)}] = min[flow(G)]$



$$flow(G) = flow^{out}(S) - flow^{in}(S)$$

$$flow(G) \leq flow^{out}(S)$$ To get the bound

$$\underline{flow(G) \leq Capacity^{out}(S)}$$

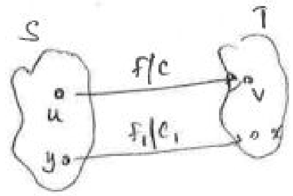
Each flow out of each edge (outgoing) from S is bounded by the capacity of that edge

largest flow = minimum Cut capacity

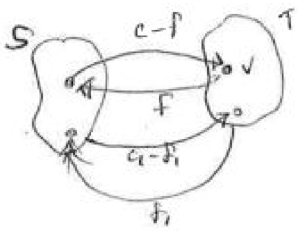
Any cut capacity is an upper bound to the flow of the network

$\Rightarrow$ The min. capacity cut will be the <u>tightest upper bound</u> you would get.

G



$G_{residual}$



$f \leq c$

$f == c$ means edge has saturated.

* What's going out is saturated to the capacity
* what's going in is 0

When we stop, we have a cut ~~tot~~ in residual G, with no outgoing edge $\Rightarrow$ Saturation of edges.

$\boxed{(\times)\cdot\ flow(G) = \text{Capacity (Cut)}}$

Question 4

1. Construct a bipartite graph such that `set1` represents the `n` students, and `set2` represents the `m` companies

2. Add a directed edge between each student and a company if and only if student nominates company and vice versa. Assign capacity of 1 to that edge.

3. Create a dummy source with links to each students in `set1`. Assign a capacity of 1 to each edge.

4. Create a dummy sink with edges from companies to sink. Assign capacities `{x1, x2, x3, .... , xm}` respectively to those edges.

5. Run max-flow algorithm over the flow network