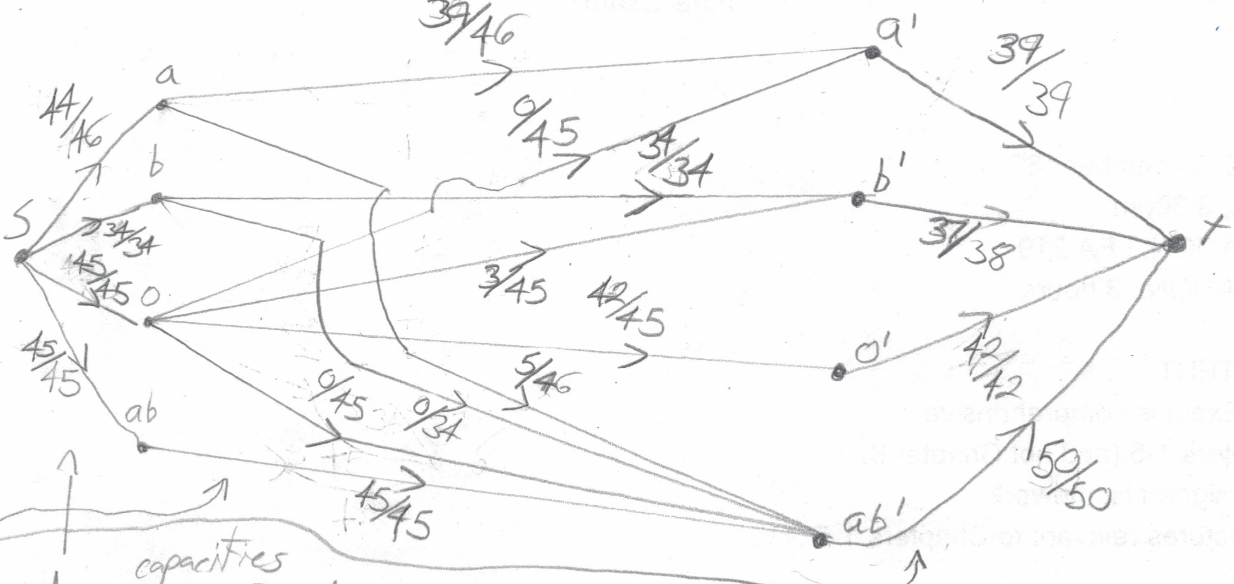
Brandon Smith

September 24, 2019

COMP 360: Algorithm Design
Assignment 1

1) a)



The edges capacities out of the source correspond to the supply of each blood type with a one-to-one correlation.

The edges out of the supply nodes show/determine which groups can receive which blood types. The capacities simply represent the supply once again.

The edges out of the demand nodes are exactly that. (One-to-one with the demands).

b)

$s \xrightarrow{0/34} b \xrightarrow{0/34} ab' \xrightarrow{0/50} t$ $+34$

$s \xrightarrow{0/46} a \xrightarrow{0/45} a' \xrightarrow{0/39} t$ $+39$

$s \xrightarrow{0/45} ab \xrightarrow{0/45} ab' \xrightarrow{34/50} t$ $+16$

$s \xrightarrow{0/45} o \xrightarrow{0/45} o' \xrightarrow{0/42} t$ $+42$

$s \xrightarrow{15/45} ab \xrightarrow{15/45} ab' \xleftarrow{34/34} b \xrightarrow{0/34} b' \xrightarrow{0/38} t$ $+29$

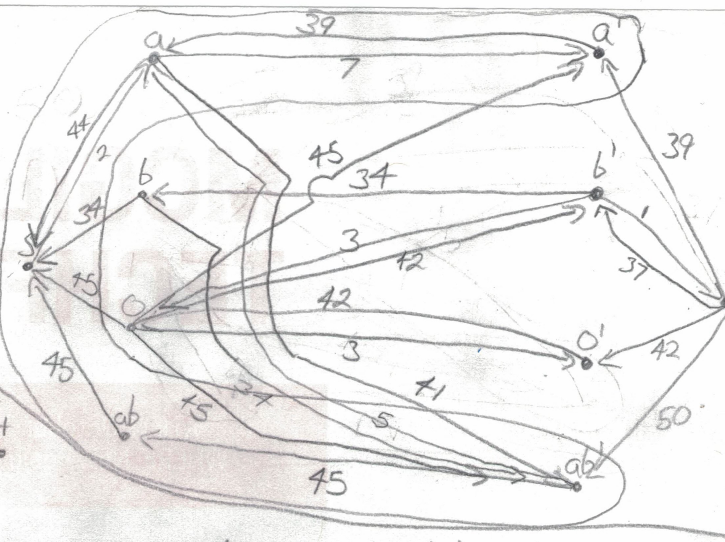
$s \xrightarrow{39/46} a \xrightarrow{0/46} ab' \xleftarrow{5/34} b \xrightarrow{29/34} b' \xrightarrow{29/38} t$ $+5$

$s \xrightarrow{42/45} o \xrightarrow{0/45} b' \xrightarrow{34/38} t$ $+3$

1c) $\{s, a, a', ab, ab'\}$

Visually the min-cut is a mess $\Rightarrow$ but if you look at the set of edges from $A \to B$, the sum of outgoing capacities is

$C_{a't} + C_{sb} + C_{so} + C_{ab't} = 34 + 34 + 45 + 50 = 168$

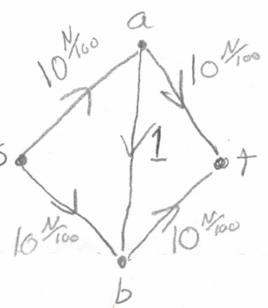


2.

$$s \xrightarrow{1} a \xrightarrow{1} b \xrightarrow{1} c \xrightarrow{1} d \xrightarrow{1} e \xrightarrow{1} f \xrightarrow{1} g \xrightarrow{1} h \xrightarrow{1} t$$

There are 9 possible distinct cuts ($m = 9 = \#\text{edges} = \#\text{cuts}$). Every one of the cuts is a min-cut because $\forall$ cuts in G the bottleneck is one.

3.



Input file: 4

0_{ss}	$10^{N/100}_{sa}$	$10^{N/100}_{sb}$	0_{st}
0_{as}	0_{aa}	1_{ab}	$10^{N/100}_{at}$
0_{bs}	0_{ba}	0_{bb}	$10^{N/100}_{bt}$
0_{ts}	0_{ta}	0_{tb}	0_{tt}

$10^K \Rightarrow K+1$ digits in base 10

So for this particular graph the input size can be simplified to

$$13 \times (7 \text{ bits}) + 4 \times \left(\frac{N}{100} + 1\right) \times (7 \text{ bits}) < N$$

And that inequality is what we want to prove

$\Rightarrow 119 + \frac{7N}{25} < N$

$\Rightarrow 119 < \frac{18N}{25}$

$\Rightarrow \frac{2975}{18} < N$

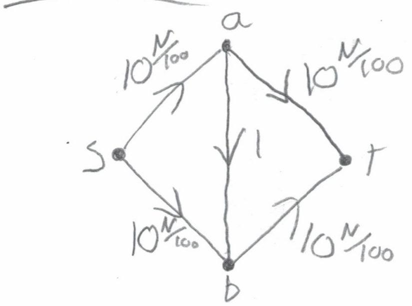
$\approx 165.2\overline{7} < N$

So the inequality holds for any $N > 166$ **AND** the bits are whole numbers so long as N is divisible by 100.

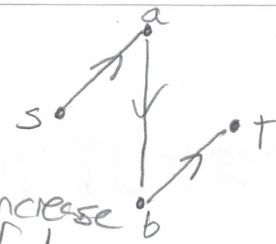
ex: $N = 1000$

$91 + 4 \times 11 \times 7 = 399 < N$ ✓

3 contd.

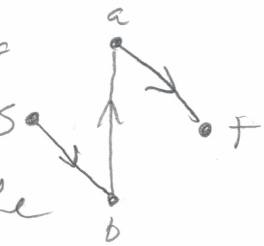


Assume initial s-t path



then the flow would increase by the bottle neck of 1.

Next assume s-t path in G_f



Once more the flow would increase by the bottle ~~neck~~ 1.

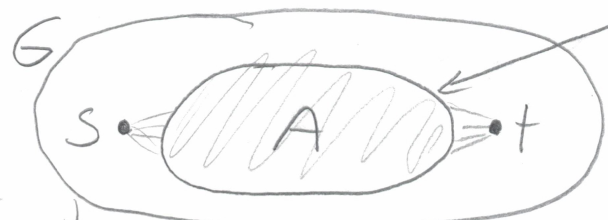
Repeat those augmenting paths for $10^{N/100}$ iterations. $10^{N/100} > 2^{N/100}$

Thus showing that worst case FF can run on exponential time.

4. Base Case:

$m = 0$ s • • t $0/2 = 0$ ✓

Assume true for m and thus A requires $m/2$ augmentations.



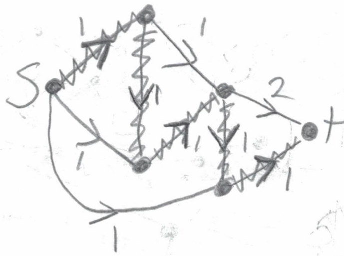
induction step:

By adding node $\{K_{n+1}\}$ (in this case n will always be $m/2$) and edges $\{sK_{n+1}, K_{n+1}t\}$ we have added an augmenting path that is independent of original graph A, and thus by adding two edges $(m+2)$ we have increased the number of augmenting paths by 1 $\left(\frac{m+2}{2} = \frac{m}{2} + 1\right)$.

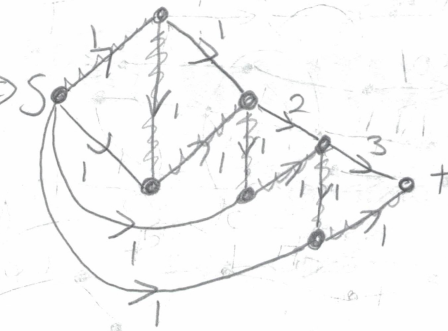
5.

max-flow = 2

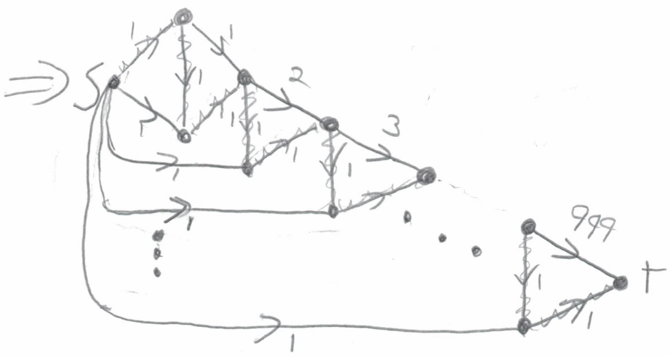
max-flow = 3



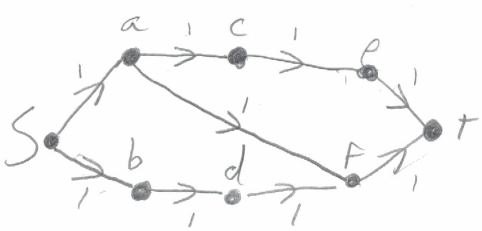
max-flow = 4



max-flow = 1000

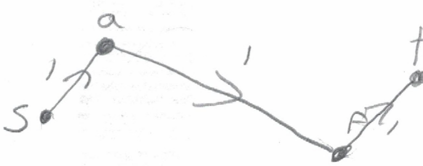


6. Counter example: (Assuming "shortest" is implying the least number of edges travelled/nodes visited)



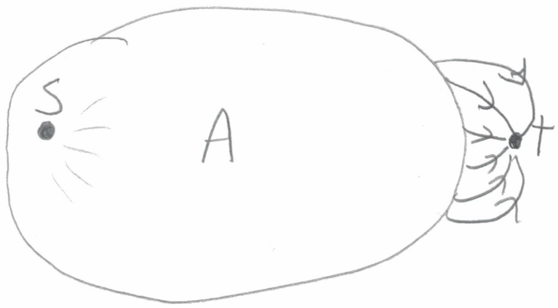
Max-flow = 2

Shortest s-t path =



Which the modified Ford-Fulkerson Algorithm would terminate immediately there-after.

7. Assuming $val(F) = F^{out}(A) - F^{in}(A)$ is true $\forall$ cuts (A, B) then we need simply look at the specific cut where $A = V \setminus \{t\}$ and $B = \{t\}$



So $Val(F) = F^{out}(A) - F^{in}(A)$

$$= \sum_{\substack{uv \in E \\ u \in A, v \in B}} F(uv) - \sum_{\substack{vu \in E \\ v \in B, u \in A}} F(vu)$$

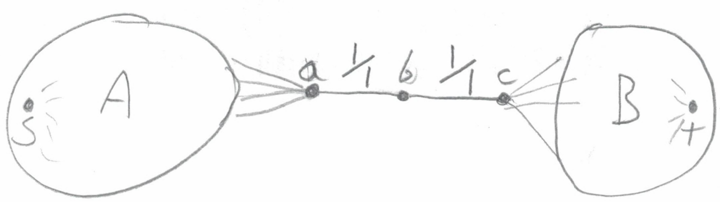
$$= \sum_{\substack{ut \in E \\ u \in A}} F(ut) - \sum_{\substack{tu \in E \\ u \in A}} F(tu)$$

$\Rightarrow$ However by the definition of a flow network the sink t only ha[s] incoming edges. So....

$$\boxed{Val(F) = \sum_{\substack{ut \in E \\ u \in A}} F(ut) - 0}$$

So coupled with the assumption that $Val(F)$ is constant for all cuts (A, B) (for a given flow F) we have proven that $Val(F)$ is equivalent to the sum of edges flowing into t.

8. The example of an edge not existing is easier so I'll show that fi[rst]



Two (or more) edges that make up the bottle-neck. You would need to increase the capacities of both edges

Given min-cut (A, B), find augmenting path from source s to $u \in A$ such that uv exists and $v \in B$ for all uv edges from A to B.
Then find augmenting path from $v \in B$ to sink t. ...