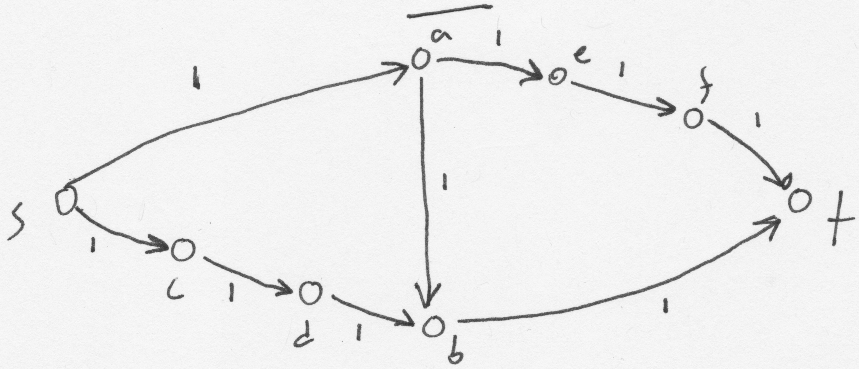
PSET 5 Wdc 22

1

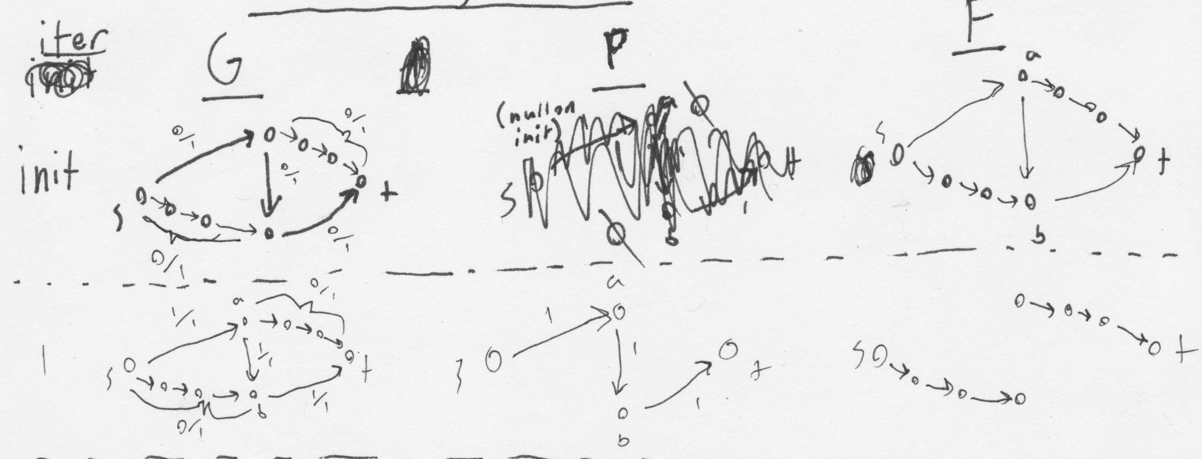
a.)



<u>Observation 1</u>: addition of nodes on some path P (say, $P = \{s, c, d, b\}$) with all $(u,v) \in P$ having capacities $\geq$ bottleneck(P) have no effect on that path's max. flow.

<u>Observation 2</u>: addition of nodes to a graph's minimum (shortest) s-t path can effect that path's state of being "shortest".

<u>dumb-algorithm</u> (no offense!)

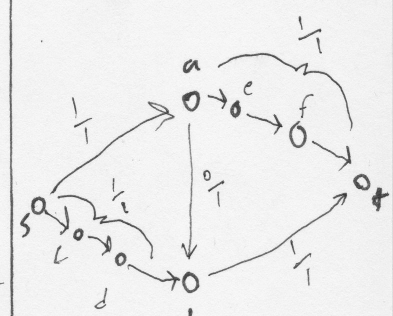


<u>HALT.</u> No $\{s\text{-}t \text{ path}\} \in (V, F)$!

$$f^{out}_{dumb}(s) = f^{in}_{dumb}(t) = \underline{1}$$

<u>Max Flow</u>

(via Ford-Fulkerson, iterations omitted for brevity)



$$f^{out}_{max}(s) = f^{in}_{max}(t) = \underline{2}$$

max flow is 2!

As you can see, the greedy algorithm produces a non-maximal flow in G.

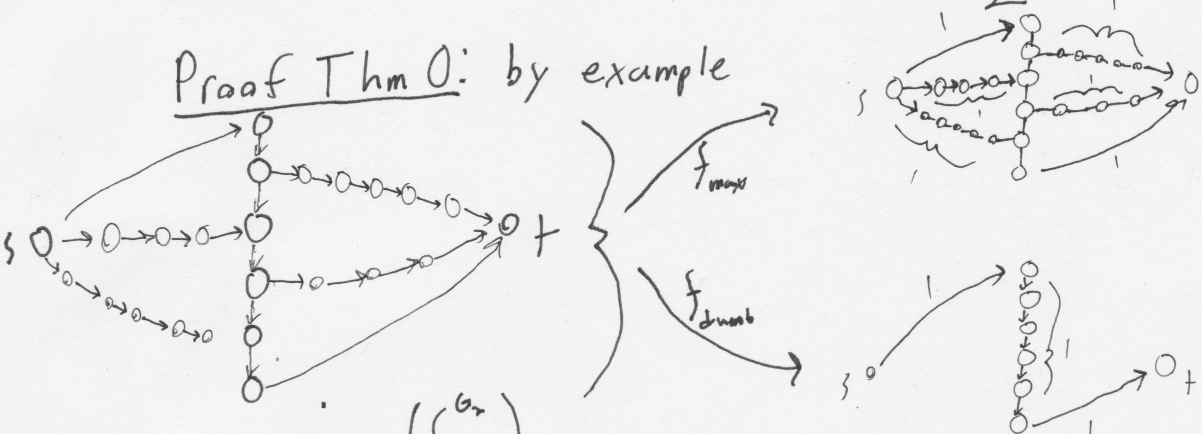
b.) we will prove that for some (or any) $k > 1, \exists G_k$ (with capacitance restrictions per the PSET) s.t. $V\left(f_{dumb}^{G_k}\right) < \dfrac{V\left(f_{max}^{G_k}\right)}{k}$ by induction.

Base Case: $k = 2$

$\boxed{c(e) = 1\ \forall e \in E}$

Thm 0: $\exists\, G_2$ s.t. $V\left(f_{dumb}^{G_2}\right) < \dfrac{V\left(f_{max}^{G_2}\right)}{2}$

Proof Thm 0: by example



$\Rightarrow V\left(f_{max}^{G_2}\right) = 3$

$\Rightarrow V\left(f_{dumb}^{G_2}\right) = 1$

$$V\left(f_{dumb}^{G_2}\right) < \frac{V\left(f_{max}^{G_2}\right)}{2} \Rightarrow 1 < \frac{3}{2} \checkmark$$

Ind. Hypothesis: $\left\{\exists\, G_k \text{ s.t. } V\left(f_{dumb}^{G_k}\right) < \dfrac{V\left(f_{max}^{G_k}\right)}{k}\right\} \rightarrow \left\{\exists\, G_{k+1} \text{ s.t. } V\left(f_{dumb}^{G_{k+1}}\right) < \dfrac{V\left(f_{max}^{G_{k+1}}\right)}{k+1}\right\}$

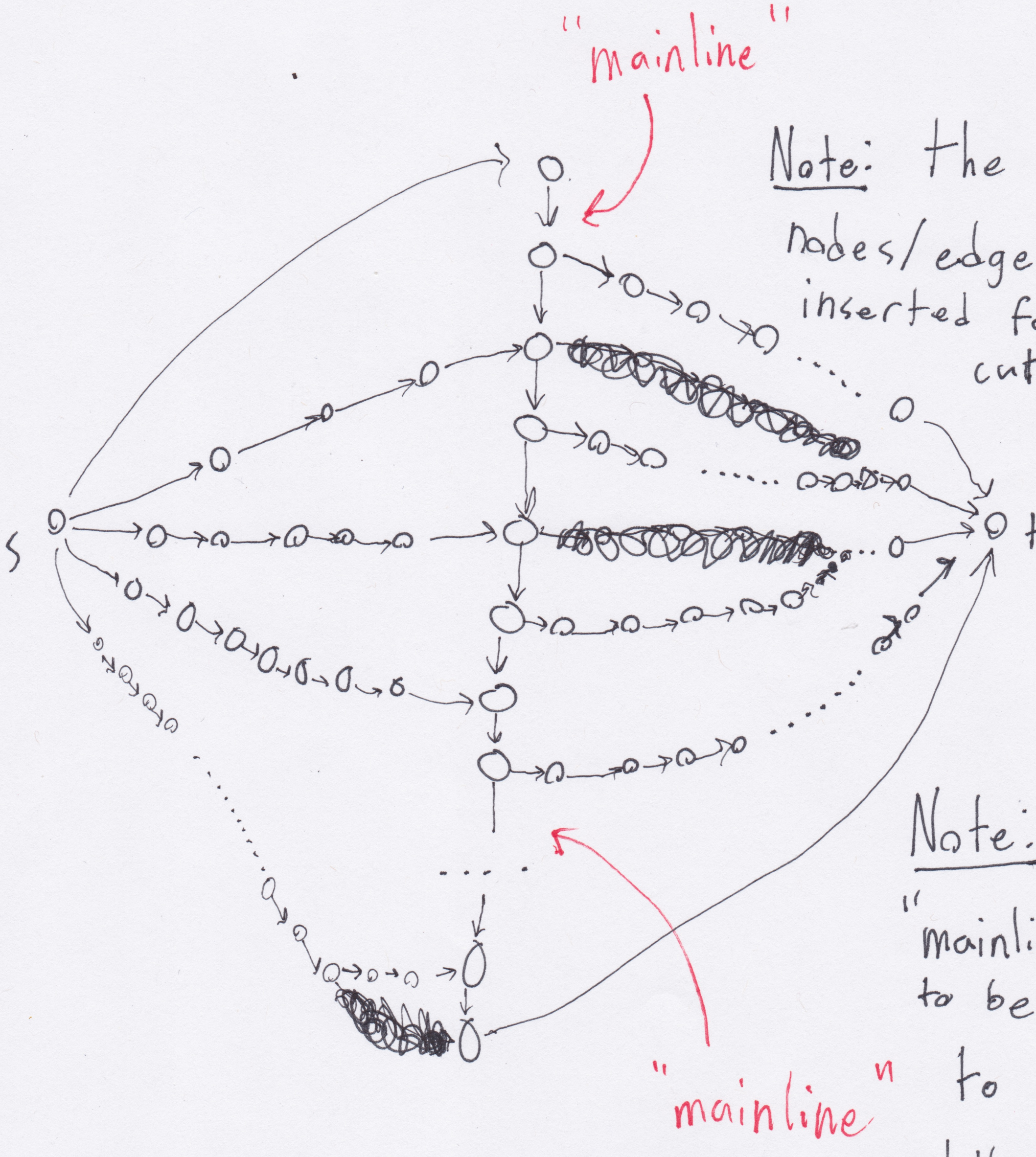
Thm 1: Inductive Hypothesis

Proof Thm 1:

Observations 1+2 are very powerful, as we saw in "Proof Thm 0". By adding interstitial nodes and edges to G_k, we can control the behavior of the shortest path search without augmenting G_k's maximal flow. This property allows us to arbitrarily, even infinitely, add interstitial nodes to influence the dumb alg.'s shortest s-t path search such that it removes all edges from F ~~which~~ through which any s-t path in G_k goes. In other words, we can manipulate the algorithm to always remove an s-t cut

1 (cont.'d)

b.) (cont.'d) from G_k on its first pass, leaving a flow of 1 in all cases. Also, we can always add more nodes to the middle portion of the s-t cut chain to increase max flow in order to make it larger than any arbitrary k (or k+1 ☺), making $\frac{v(f^{G_k}_{max})}{k} > 1$ for any k. It is for this reason (arbitrary k) that "Thm 1" holds ∎



<u>Note:</u> the number of interstitial nodes/edges which need to be inserted for the "mainline" s-t cut to be severed varies depending on distance from t.

<u>Note:</u> the number of "mainline" nodes which need to be inserted for "Thm 1" to hold is $2(k+1)$ for LHS, $2(k+2)$ for RHS of "Thm 1".