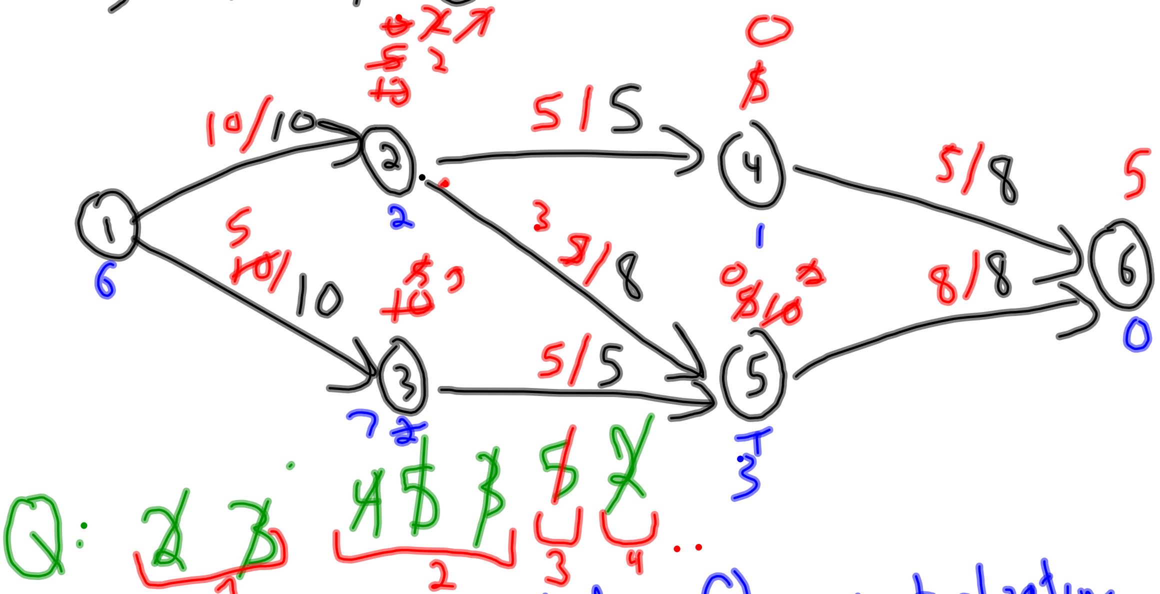
Choosing which vertex to call Discharge(v)

1) FIFO



phase 1: nodes added to Q in initializing

phase i: nodes added during phase i-1

Mar 3-4:09 PM

Claim # of phases is $O(n^2)$.

Pf $\Phi = \max \{ d(v) : \underset{v \text{ on } Q}{e(v) > 0} \}$

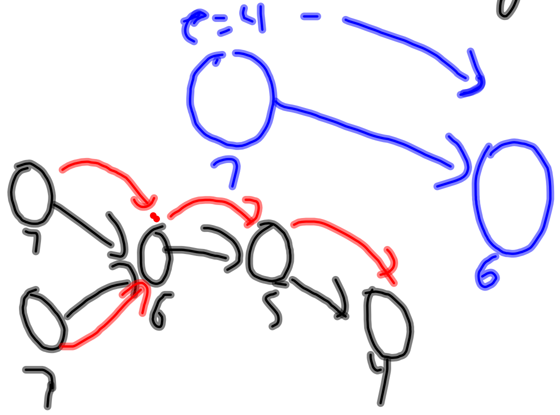
~~7~~ 5 ~~4~~ 8 ~~6~~ ~~3~~

Look at Φ during a phase

Case 1: A relabeling occurs during the phase

Happens $\leq 2n^2$ times. Total increase in Φ in these phases $\leq 2n^2$.

Case 2: No relabeling occurs during the phase



v_6 v_{10} v_{12} active

$d(v_6)=8 \to 7$, $d(v_{10})= 12 \to 11$, $d(v_{12}) = 7 \to 6$

vertices w/ $e(v) > 0$ & $d(v) = \Phi$ at start of phase, do not have excess at end of phase.

No new excess at Φ or higher

$\therefore$ Φ decreases by at least 1.

$\Phi_{init} \le n$

$\Phi_{final} = 0$

total increase in $\Phi \le 2n^2$

total decrease in $\Phi \le 2n^2 + n - 0 \le 3n^2$

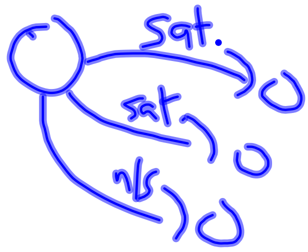
each phase w/o a relabeling decrease Φ by ≥ 1.

$\therefore \le 3n^2$ phases w/o a relabeling.

$\therefore \le 5n^2$ phases overall.

*# n.s. pushes | phase $\leq n$

n.s. pushes per Discharge(v) ≤ 1



calls to Discharge per phase $\leq n$.

$\therefore$ # of n.s. pushes $= O(n^3)$

Mar 3-4:43 PM

nm sat. pushes
nm overhead
nm relabeling
n^3 n.s. pushes

$O(n^3)$ time.

$n^2 m$

Mar 3-4:46 PM

Excess scaling

$e_{max} \equiv \max\{e(v)\}$

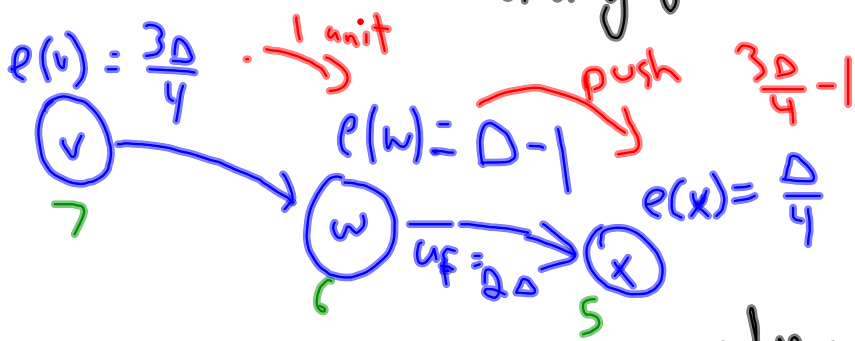
Only push flow from vertices w/ $e(v) \approx e_{max}$
gradually decrease e_{max}

Δ an upper bound on e_{max}

A node w/ $e(v) \geq \frac{\Delta}{2}$ has large excess
$e(v) < \frac{\Delta}{2}$ " small excess

Invariants to maintain:

1) only push flow from nodes w/ large excess
2) never let $e(v) > \Delta$ for any v



Node selection rule: Choose, among nodes w/ large excess, the one w/ min dist. label

$\Rightarrow$ push from nodes of large excess to small excess

Excess scaling alg

- Preprocess
- $\Delta = 2^{\lceil \lg u \rceil}$
- while $(\Delta \geq 1)$

while some node has large excess

right dis. → · choose v to be the min dist. labelled node of large excess

(w = current-edge(v)

· if push(v,w) is applicable

push $\delta = \min(e(v), u_f(v,w), \Delta - e(w))$

else if not at end of list advance w

else relabel(v)

$\Delta = \Delta / 2$

Lemma

1) Each n.s. push sends $\geq \frac{\Delta}{2}$ units

2) No excess ever exceeds Δ.

Pf

2 ✓

1) Sending $\min e(v), \Delta - e(w)$

but $e(v) \geq \frac{\Delta}{2}$, $e(w) < \frac{\Delta}{2}$,

$\Delta - e(w) \geq \frac{\Delta}{2}$.

Lemma $O(n^2)$ n.s. pushes per Δ-scaling phase.

$e_{max} \cdot \le \Delta \ldots \rightsquigarrow e_{max} \le \Delta/2$

Pf $\Phi = \sum_v e(v) \cdot d(v) / \Delta$

At beginning of a scaling phase

$$\Phi \le \sum_v d(v) \frac{e(v)}{\Delta}$$

$$\le \sum_v 2n \, (1) \le 2n^2.$$

At end of a scaling phase

$$\Phi \ge 0$$

$$\Phi = \sum_v e(v)d(v)/\Delta = \sum_v d(v)\frac{e(v)}{\Delta}$$

Relabel: Φ increases. If $d(v)$ increases by α

Φ increases by $\leq \alpha$. from relabelings

$\Rightarrow$ Total increase in $\Phi \leq 2n^2$

Push: sat. push: decreases Φ, does not increase Φ

(1 unit from $d(v) = 2$, $d(w) = 1$

$\Delta = 10^6$, decrease in $\Phi = \frac{1}{10^6}$

non-sat push: decreases Φ

n.s. push (v,w) sends $\delta \geq \frac{\Delta}{2}$ units

$$\text{decrease in } \Phi = \frac{\delta\,(d(v)-d(w))}{\Delta} \geq \frac{(\Delta/2)\,(1)}{\Delta} = \frac{1}{2}$$

$$\#\text{ n.s. pushes} \leq \frac{2n^2 - 0 + 2n^2}{1/2} = 8n^2$$

Mar 5-4:19 PM

n^2 relabels

nm time relabeling

nm sat. pushes

$\lceil \lg U \rceil$ scaling phases

each one does $O(n^2)$ n.s. pushes

$O(n^2 \lg U)$ n.s. pushes

$m \approx n^2$ n^3

$m \approx n$ $n^2 \lg U$

$$\text{Time} = O(nm + n^2 \lg U)$$

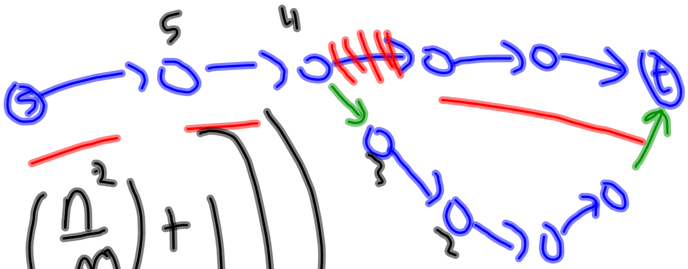
FIFO

Exsess Scaling

Highest label - in practice is best

<u>Data Structures</u>

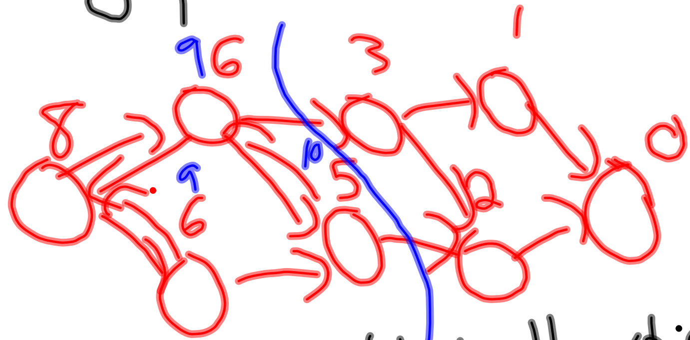
Dynamic Tree - stores paths



$$O\left(nm\left[\lg\left(\frac{n^2}{m}\right)+1\right]\right)$$

Push/Relabel works well in practice
as long as

- reasonable data structures
- gap heuristic



m time

Every 10m steps

- periodically relabel all vertices via BFS.

increase running time by ≤10%
potentially decrease by alot.