Input $(G=(V,E), s, t)$

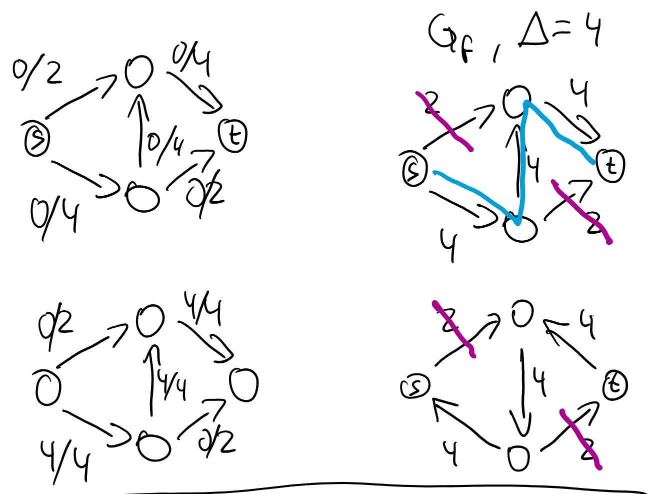
$f_e = 0 \quad \forall e \in E$

$\Delta := \max_{(u,v)\in E} c_{uv}$ rounded to closest power of 2

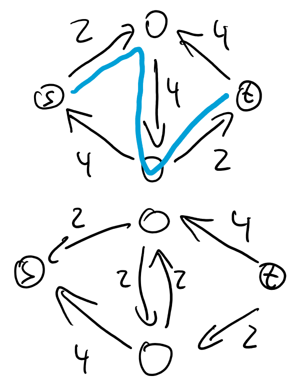
while $\Delta \geq 1$ ← Phase for certain Δ

- while True
  - construct G_f
  - delete all edges from G_f with capacity $< \Delta$ ← capacity in G_f
  - if path $s \rightsquigarrow t$ exists in G_f
    - augment f by path
  - else
    - break
- $\Delta = \Delta / 2$

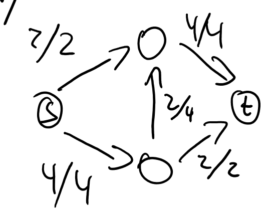
G_f, $\Delta = 4$



$\Delta = 2$



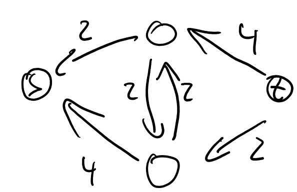
flow/capacity



max flow

$O(|E|^2 \cdot \log C)$ ← $\max_{(u,v)\in E} c_{uv}$

$\Delta = 1$



terminate

Correctness of algorithm:

In last iteration $\Delta = 1$

so no edge is deleted in G_f

The algo terminates if there is no path in G_f

Proven last week: this implies flow is maximum.

Time: $O(|E| \cdot \#\text{iterations}) \rightarrow O(|E|^2 \log C)$

Claim 1) # iterations within phase $\Delta \in O(|E|)$

$\Rightarrow$ total # iterations over all phases is $O(|E| \log C)$

because there are $O(\log C)$ many phases.

Previous $O(F \cdot |E|)$

$O(|V| C \, |E|)$

↑ k bit integers then $C \sim 2^k$

$O(|V| \cdot 2^k \, |E|)$

Claim 2) At the end of one phase we have

Claim 2) At the end of one phase we have

$$\underset{\substack{\uparrow\\ \text{computed by algo so far}}}{\text{size}(f)} \geq \underset{\substack{\uparrow\\ \text{max flow for graph } G}}{\text{size}(f^*)} - \underline{|E|\cdot\Delta\cdot 2}$$

At the end of a phase, we decrease Delta by factor 2. So at the start of a new phase (which has a new smaller Delta), we have size(f) >= size (f*)-|E| Delta 4

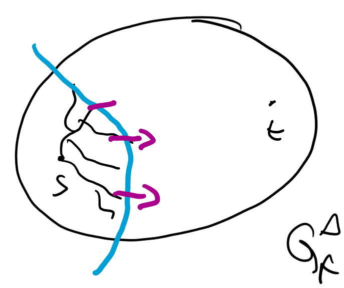
Proof Claim 1) In each iteration, the flow increases by Δ but we are at most $\underline{|E|\cdot\Delta\cdot 4}$ far away from the size of max flow so we can increase the flow at most $\frac{|E|\cdot\Delta\cdot 4}{\Delta} = 4|E|$ times.

Proof Claim 2) When a phase terminates, there is no st-path in G_f^Δ

Let $W \subseteq V$ be the vertices we can reach from s in G_f^Δ

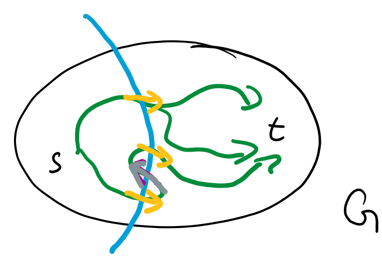
$$\underbrace{\text{size}(f^*)}_{\leq \delta(W)} - \text{size}(f) \leq \delta(W) - \text{size}(f)$$

$$= \sum_{\substack{(u,v)\in E\\ u\in W\\ v\notin W}} c_{uv} - \Big(\sum_{\substack{(u,v)\in E\\ u\in W\\ v\notin W}} f_{uv} - \sum_{\substack{(u,v)\in E\\ u\notin W\\ v\in W}} f_{uv} \Big)$$



purple edges have capacity $< \Delta$

$$= \sum_{\substack{(u,v)\in E\\ u\in W\\ v\notin W}} \overbrace{(c_{uv} - f_{uv})}^{\leq\Delta} + \sum_{\substack{(u,v)\in E\\ u\notin W\\ v\in W}} \overbrace{f_{uv}}^{\leq\Delta}$$



$$\leq |E|\cdot\Delta + |E|\cdot\Delta = 2\cdot|E|\cdot\Delta$$

Linear Program

Bread A needs 2 flour, 1 egg, sells for \$3

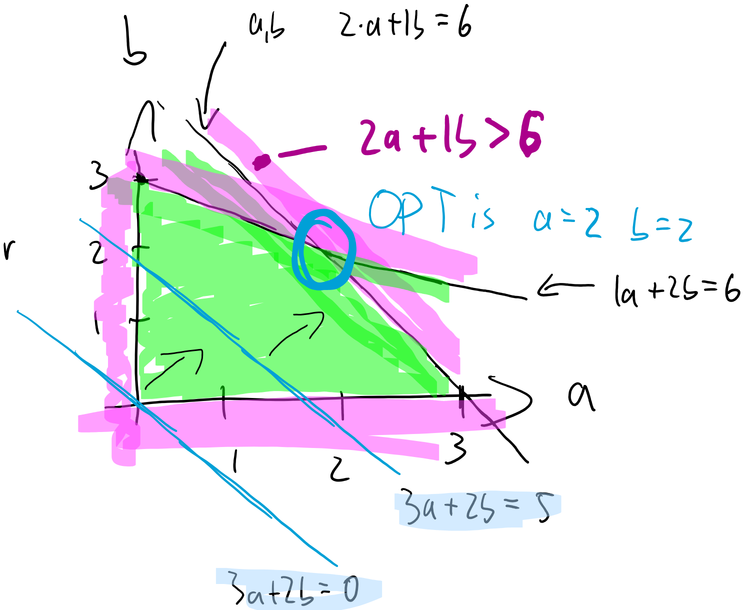
Bread B needs 1 flour, 2 egg, sells for \$2

Bread B needs 1 flour, 2 egg, sells for \$2

We have 6 flour 6 egg.

Q: What should we bake? (We are allowed to pick fractional values)

$a=3, b=0 \Rightarrow \$9$ $\qquad$ $a=2, b=2 \Rightarrow \$10$



$$\max \ 3 \cdot a + 2 \cdot b$$

$2 \cdot a + 1 \cdot b \le 6$ // usage of flour

$1 \cdot a + 2 \cdot b \le 6$ // usage of egg

$a \ge 0$

$b \ge 0$

Def: Linear Program

$A \in \mathbb{R}^{n \times d}$ $n \times d$ $n \ge d$

$c \in \mathbb{R}^d$ d-dim

$b \in \mathbb{R}^n$ n-dim

$$\max \ c^T x$$

$$Ax \le b \leftarrow (Ax)_i \le b_i \ \forall i$$

$$\max \ 3 \cdot a + 2 \cdot b$$

$2 \cdot a + 1 \cdot b \le 6$

$1 \cdot a + 2 \cdot b \le 6$

$$\boxed{\begin{aligned} a &\ge 0 \\ b &\ge 0 \end{aligned}}$$

$\Leftrightarrow$ $-a \le 0$

$-b \le 0$

$$\max \begin{pmatrix} 3 \\ 2 \end{pmatrix}^T \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \le \begin{pmatrix} 6 \\ 6 \\ 0 \\ 0 \end{pmatrix}$$

Remark: both $\le$ and $\ge$ are equivalent, just multiply by -1

Remark: both $\leq$ and $\geq$ are equivalent, just multiply by -1

$$2a + 5b = 6 \longrightarrow \begin{array}{l} 2a + 5b \leq 6 \\ 2a + 5b \geq 6 \end{array} \longrightarrow \begin{array}{l} 2a + 5b \leq 6 \\ -2a - 5b \leq -6 \end{array}$$