ADM Ex. Session:

10.01.14

Announcements:

- Send your PA1 till Monday, 8:00
- choose date for presentation (lists to sign in)
- next week: • wednesday: no lecture, no tutorial because of present. of PA1
  - • Thursday/Friday: Lecture
- evaluation sheets

max Flow: • FF/ED, Impl. details, Dinics/Dinitz

given: Network $(\underbrace{G}_{\text{graph}}, \underbrace{u}_{\substack{\text{capacities}\\ u: E \to \mathbb{Q} / \mathbb{N}}}, \underbrace{s, t}_{\text{source/sink}})$

sol.: flow $f: E \to \mathbb{Q}_0^+ / \mathbb{N}_0$

s.t.:

feasibility

(1) flow conservation

$$\forall\ v \neq s,t \qquad \underbrace{\sum_{e \in \delta^-(v)} f_e}_{\text{incoming flow}} = \underbrace{\sum_{e \in \delta^+(v)} f_e}_{\text{outgoing flow}}$$

(2) capacity: $\forall e \in E: \quad 0 \leq f_e \leq u(e)$

optim-ality (3) Maximality: (wrt. $v(f)$)

$$v(f) := \sum_{e \in \delta^+(s)} f_e - \sum_{e \in \delta^-(s)} f_e$$

$$= \sum_{e \in \delta^-(t)} f_e - \sum_{e \in \delta^+(t)} f_e$$

Max-flow mincut theorem

"value of a max. s-t flow is equal to the value of an s-t cut."

$$\max_{\substack{f \text{ is an} \\ \text{s-t flow}}} v(f) = \min_{\substack{A \text{ is an} \\ \text{s-t cut}}} \operatorname{cap}(A)$$

$\cdot\ \operatorname{cap}(A) := \sum_{\substack{e=(v,w), \\ v \in A \\ w \notin A}} u_e$

Ford-Fulkerson:

WHILE $\exists$ f-augmenting s-t path in G_f : (residual network)

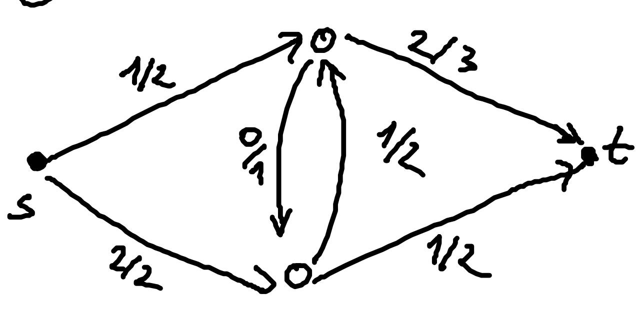
augment f along path with value $\delta := \min_{\substack{(i,j) \in E(G_f) \\ \in \text{path}}} \{r_{ij}\}$

Implementation

- often only residual capacity needed

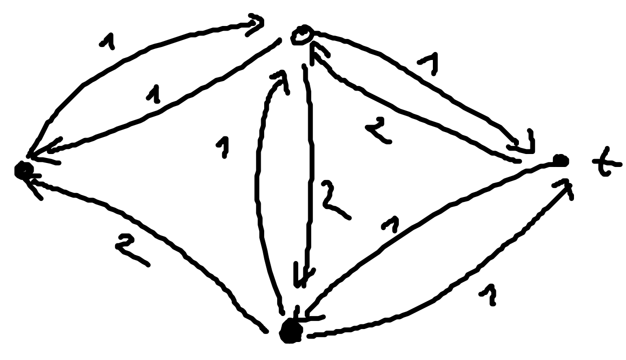
=> flow can calculated from res. cap. at the end of an algorithm

G



~>

G_f



f_e/u_e

combine parallel edges

given path p in G_f :

augment flow, $\delta := \min_{(i,j) \in p} \{r_{ij}\}$

$$\forall (i,j) \in P: \begin{cases} r_{ij} := r_{ij} - \delta & \text{Forward edge} \\ r_{ji} = r_{ji} + \delta & \text{Backward edge} \end{cases}$$

→ fast access to (j,i), $O(1)$

→ edge knows its anti parallel edge

How to find the flow value, using all residual cap.,
(many flows might result in same res. network)

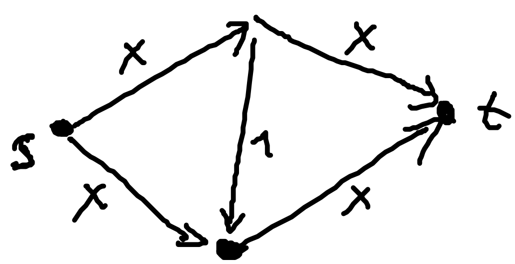
$$r_{ij} = u_{ij} - f_{ij} + f_{ji}$$

$$\Leftrightarrow \quad \underbrace{u_{ij} - r_{ij}}_{\geq 0} = f_{ij} - f_{ji}$$

$$\Rightarrow \left.\begin{aligned} f_{ij} &:= u_{ij} - r_{ij} \\ f_{ji} &:= 0 \end{aligned}\right\} \quad \text{possible choice for flow } f_{ij},\ f_{ji}$$

$$< 0 \Rightarrow \left.\begin{aligned} f_{ji} &:= r_{ij} - u_{ij} \\ f_{ij} &:= 0 \end{aligned}\right\} \quad \text{—— '' ——}$$

Ford Fulkerson: $\boxed{O(m \cdot v(f))}$ ⇝ not polyn. in $|E| + |V| = n+m$



Edmonds Karp:
- choose shortest s-t path (# of edges)
- done by BFS in G_f
- # of edges along increases weakly monotone

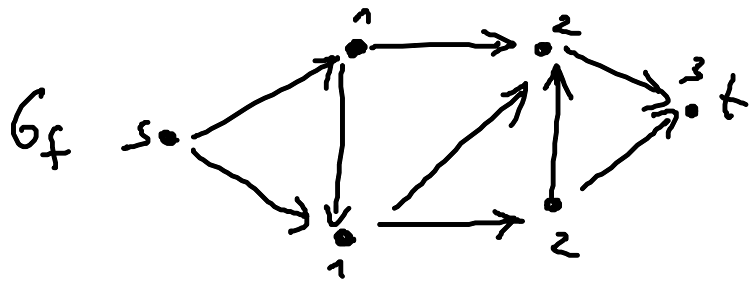
⇝ $\frac{m \cdot n}{2}$ iterations

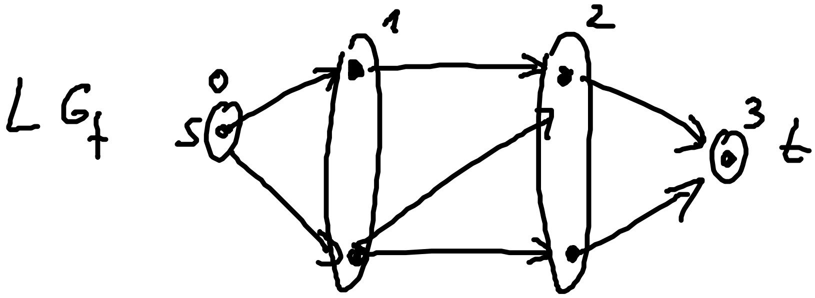
$$\Rightarrow O\left(m \cdot \frac{m \cdot n}{2}\right) = \boxed{O(m^2 n)}$$

Improvement possible: augmenting all paths with min. # of edge at the same time

⇝ ≤ n−1 iterations, length grows at least +1 in a single iteration

use **Levelgraph**: LG_f of G_f
(subgraph of G_f, using only edges which are part of min. s−t paths)





Def: blocking Flow: flow in G_f, s.th. every path with min # edges contains at least one saturated edge

Calc. of a block. Flow: (Dinic)

1. constr. LG_f of G_f $O(m)$
2. DFS finds an s−t path $O(n)$
3. augm. flow along p

≤m times

- less than m edges to saturate $\to O(n \cdot m)$ to find blocking flow
- strongly monotone path length for f augm. s-t paths
  $\to \leq n-1$ blocking flows to find

$\Rightarrow \quad O(n \cdot n \cdot m) = \boxed{O(n^2 \cdot m)}$ for Dinic

Malhotra, Kumar, Maheshwari (1974) (MKM)

<u>Def:</u> flow potential of a vertex v:

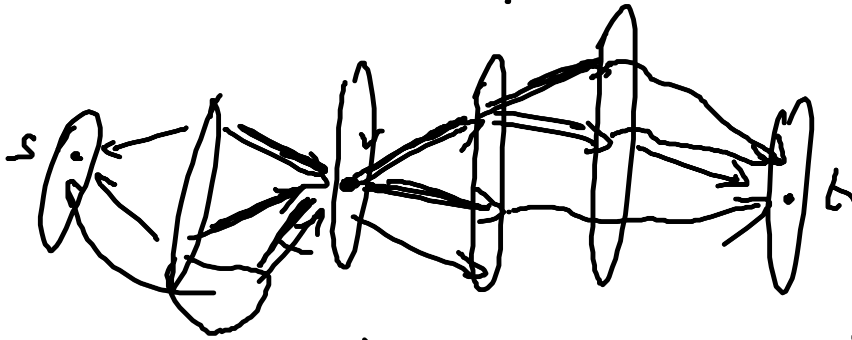
$$
p_f(v) := \begin{cases} \min\left(\sum_{e \in \delta^+(v)} u_e - f_e, \sum_{e \in \delta^-(v)} u_e - f_e \right) & v \neq s,t \\ \sum_{e \in \delta^+} u_e - f_e & v = s \\ \sum_{e \in \delta^-(v)} u_e - f_e & v = t \end{cases}
$$

Idea: "how much flow can enter or leave vertex v"
calculation in $O(m)$

<u>Algorithm (finding a blocking flow)</u>

1. find $v \in V$ with min. potential $\quad O(n)$
2. send $p_f(v)$ flow along edges leaving/entering v $\quad O(n)$
   "distribute" flow backwards + forwards

③ delete vertex v from LG_f



⇝ block flow in $O(n^2)$ ⇒ max flow in $\boxed{O(n^3)}$ MKM

<u>Jim Orlin</u> (2012): max flow in $O(n \cdot m)$

$m \in O(n^{14/15-\varepsilon})$ ⇝ solvable in $O(n\,m)$

$m > n^{1+\varepsilon}$ ⇝ — " — " — via King, Rao, Tarjan

- Max Flow ≙ Min Cut

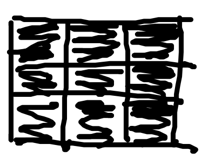
  what about max. cut in G_u? ⇝ no polyn. also.

<u>Practical Usage of min cuts:</u>

picture

segmentation

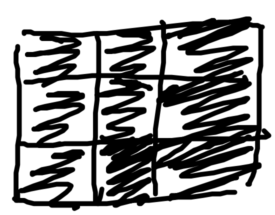
- use minimal cuts to find „good" segmentation



→

→

segmentation



- big edge weight for similar colors
- small edge weight for distinct colors