Shortest Augmenting Path Algorithm

res. graph
$d(v)$ distance from $v$ to $t$

- $d(s) = 1$
- $d(t) = 2$
- $d(s) = 4$
- $d(s) = 5$
- $d(s) = 9$
Lemma: For every $v$, $f(v)$ never decreases.

Pf:
- $f$ dist. before an aug. path
- $f'$ dist. after

Suppose some $f(v)$ decreases, let $v$ be the one with minimum $f(v)$ to decrease.

If $(v, w)$ is in $G_f$ but not $G_{f'}$, then we sent flow on $(w, v)$ in the augmentation. Then $\delta'(w) = 6$, contradicts $\delta'(w) = 3$. Contradiction.
- $d(v)$ never decreases
- For each $v$, $d(v)$ increases $\leq n$ times.
- Total # of times that any $d(v)$ increases $\leq n^2$.

If a push sends $U_f(v,w)$ flow from $v$ to $w$, we call it saturating.
Facts

1) Between any 2 saturations of $(v,w)$ or $(w,v)$, either $d(v)$ or $d(w)$ must increase by 2.

$\#$ saturations of $(v,w) or (w,v) \leq n$

2) Every aug. path saturates at least 1 residual edge.

$\#$ aug paths $\leq \#$ saturations

$\leq \mathcal{O}(nm)$

$= \mathcal{O}(nm)$

R.T is $\mathcal{O}(nm^2)$

Edmonds, Karp
72
\( O(n^2 m) \) a little careful

Blocking Flows - send along "all" shortest paths simultaneously

Repeat \( n \) times

Data Structures
Scaling of capacities

Consider highest order bit 1. Solve

Update add a bit to $u$ double the flows or doubles $u + 1$
1. Correctly compute a max flow
   - always feasible
   - always sending flow on a residual path in $G_f$

2. Running time $\log U$ iterations of scaling

   Each iteration is a sequence of $x$ augmenting paths

   Bound how much flow gets sent in an iteration

   \[
   f = \min cut
   \]

\[
uf(vw) \leq u(vw) + 1
\]

\[
u_f(S, T) \leq m.
\]
\[ \exists \text{ a cut in } G_f \text{ w/ } u_f(\delta(T)) \leq m \]

\[ \therefore \text{ total flow we can send in } G_f \leq m \]

\[ \therefore \text{ at most } m \text{ iterations of aug. paths.} \]

\[ \mathcal{O}(m^2 \lg n) \text{ time} \]