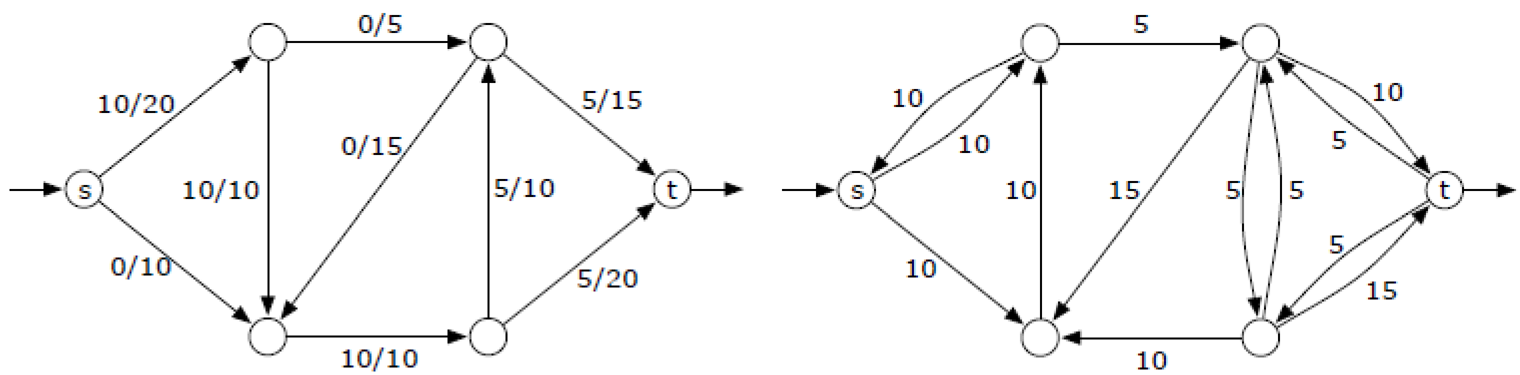
CS314 – Recap of flows

Note Title 10/25/2013

- HW due ~~today~~ today
- Next HW – written, due in 1 week

The algorithm: all about the residual graph



A flow f in a weighted graph G and the corresponding residual graph G_f.

Analysis:

How many paths could you have to find in the residual graph?

You saw: $O(m|f|)$

But can we ~~do~~ analyze better?

Q: Which path in the residual should we take?

Ideas:

- largest bottleneck path
- take path w/ fewest edge

Edmonds - Karp: take a "fat" pipe
(largest bottleneck edge)

A greedy approach!

$O(m^2 \log n \log |f|)$

Dinits (Edmonds-Karp):

Choose a short path:

↳ BFS : $O(m)$

Bounding how levels increase in G_f gives:

$$O(n \cdot m^2)$$

Best Possible: Unknown!

Technique	Direct	With dynamic trees	Sources
Blocking flow	$O(V^3)$	$O(VE \log V)$	[Dinits; Sleator and Tarjan]
Network simplex	$O(V^2E)$	$O(VE \log V)$	[Dantzig; Goldfarb and Hao; Goldberg, Grigoriadis, and Tarjan]
Push-relabel (generic)	$O(V^2E)$	—	[Goldberg and Tarjan]
Push-relabel (FIFO)	$O(V^3)$	$O(V^2 \log(V^2/E))$	[Goldberg and Tarjan]
Push-relabel (highest label)	$O(V^2\sqrt{E})$	—	[Cheriyan and Maheshwari; Tunçel]
Pseudoflow	$O(V^2E)$	$O(VE \log V)$	[Hochbaum]

(plus many more...)