CS 4800: Algorithms & Data

Lecture 20

April 4, 2017

Optimality of Ford-Fulkerson

Showed:

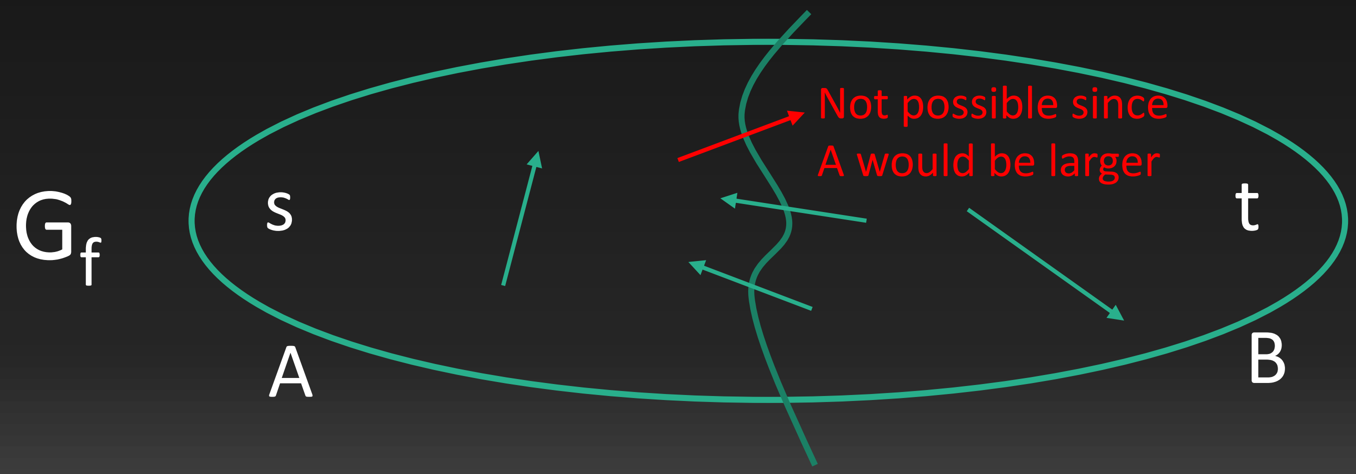
- For all flow f and cut (A,B), $|f| \leq cap(A, B)$

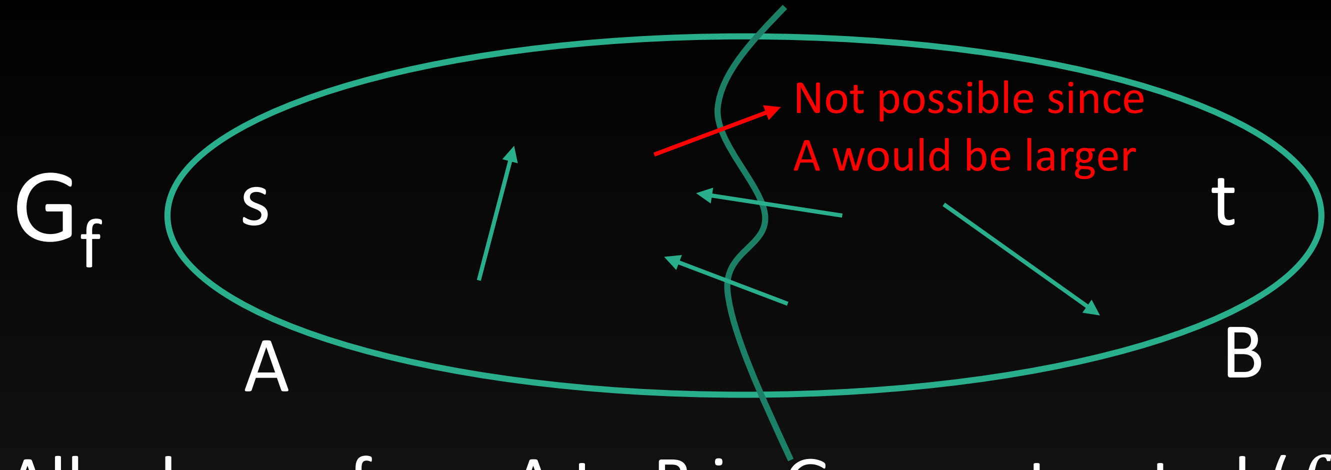
3 equivalent statements:

- f is maximum flow
- There is s-t cut (A,B) such that |f| = cap(A,B)
- There is no augmenting path in G_f

No augmenting path implies |f|=cap(A,B) for some A,B

- Define A = {v reachable from s in G_f}, $B = V \setminus A$
- s is reachable from s to $s \in A$
- t is not reachable from s so $t \notin A$





- All edges e from A to B in G are saturated ($f(e) = c(e)$) since e goes backward in G_f
- All edges e from B to A in G are not used since there is no backward edge from A to B ($f(e) = 0$)
- Thus,

$$|f| = \sum_{u \in A, v \in B} f(u,v) - f(v,u)$$

0

- |f|=cap(A,B)

Max-flow/min-cut theorem

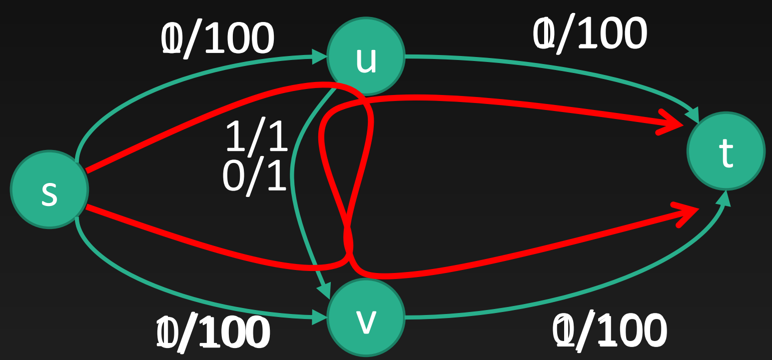
- Maximum flow = minimum cut

Computing min cut

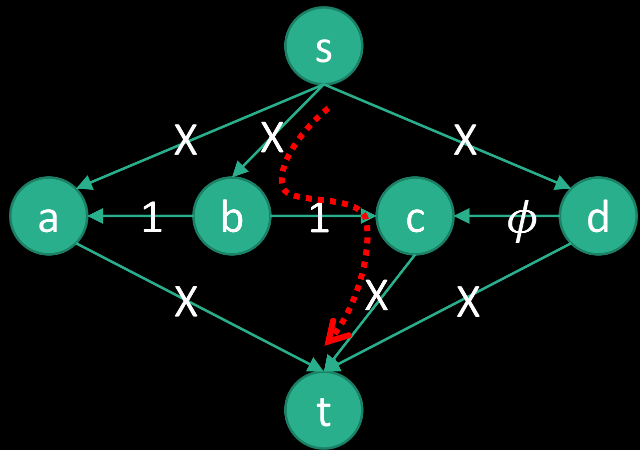
- Given max flow, can compute min cut in O(V+E) time
- Use BFS to find all vertices reachable from s in G_f
- Let A ={vertices reachable from s in G_f}
- The cut $(A, V \setminus A)$ has $\mathrm{cap}(A, V \setminus A) = |f|$ and hence is minimum

How fast is Ford-Fulkerson?

G

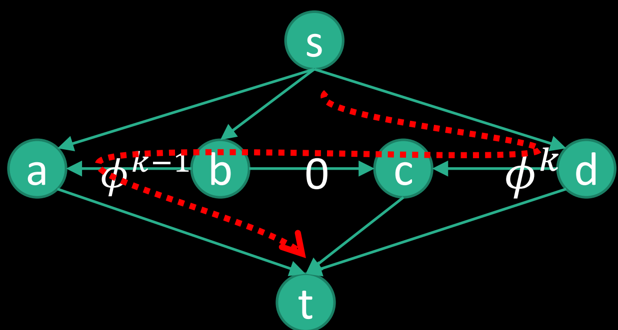


As much time as $E \cdot |f^*|$

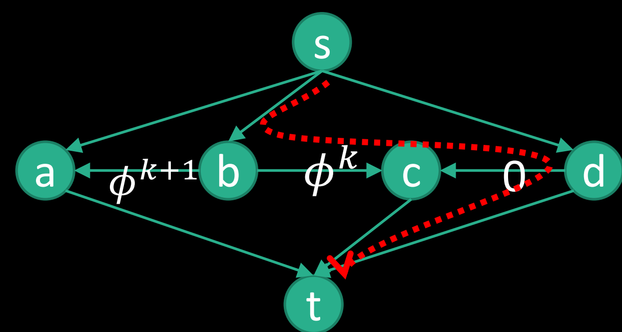


- $\phi = (\sqrt{5} - 1)/2$ so $1 - \phi = \phi^2$
- Max flow = 2X + 1
- After 1st augmentation, residual capacities of horizontal edges are 1, 0, ϕ

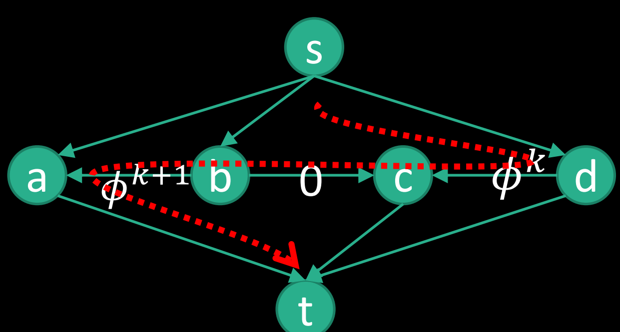
Suppose inductively that residual capacities are $\phi^{k-1}, 0, \phi^k$



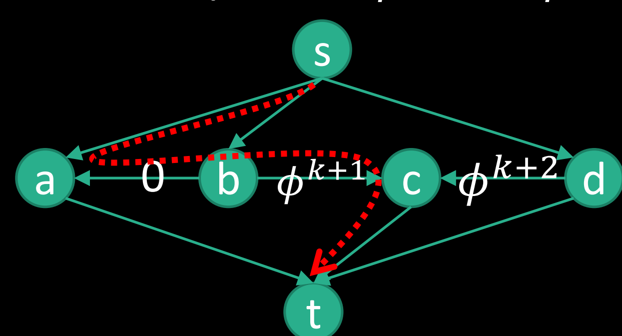
New capacities $\phi^{k+1}, \phi^k, 0$



New capacities $\phi^{k+1}, 0, \phi^k$



New capacities $0, \phi^{k+1}, \phi^{k+2}$



New capacities $\phi^{k+1}, 0, \phi^{k+2}$

Total flow

$+\phi^k \quad +\phi^k$

$+\phi^{k+1} \quad +\phi^{k+1}$

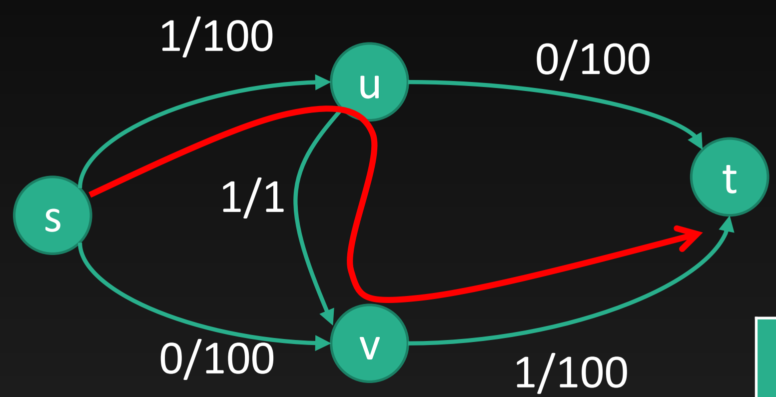
Total flow converges to

$$1 + 2\sum_{k=1}^{\infty} \phi^k = 4 + \sqrt{5}$$

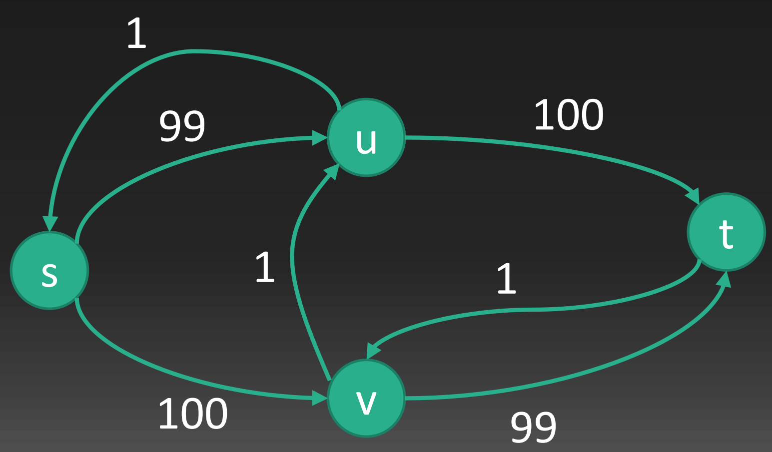
Dinitz/Edmonds-Karp

- Choose augmenting path with fewest edges
- Use BFS on G_f to find augmenting path

- G_i : residual graph after i augmentation steps
- $level_i(v)$: unweighted shortest path distance from s to v after i augmentation steps



Vertex	s	u	v	t
Level	0	1	1	2



- Edge (u,s) appears AFTER augmentation on (s,u)
- Edge (u,v) disappears

Level increases monotonically

Lemma. $level_i(v) \leq level_{i+1}(v)$ for all v, i.

Proof. Fix i. We prove by induction on the value of $level_{i+1}(v)$.

In base case, $level_{i+1}(v) = 0$. It must be $v = s$ and $level_i(s) = 0$.

In inductive case, assume lemma is true for all v with $level_{i+1}(v) < k$.

Will prove lemma for v with $level_{i+1}(v) = k$.

Let $s \rightarrow \cdots \rightarrow u \rightarrow v$ be shortest path from s to v in G_{i+1}

This path is shortest so $level_{i+1}(u) = level_{i+1}(v) - 1 = \mathrm{k} - 1$.

By induction, $level_i(u) \leq level_{i+1}(u)$.

1) If (u,v) is an edge in G_i then $level_i(v) \leq level_i(u) + 1 \leq k$.

2) If (u,v) is not an edge in G_i then (v,u) is an edge in i+1st augmenting path.

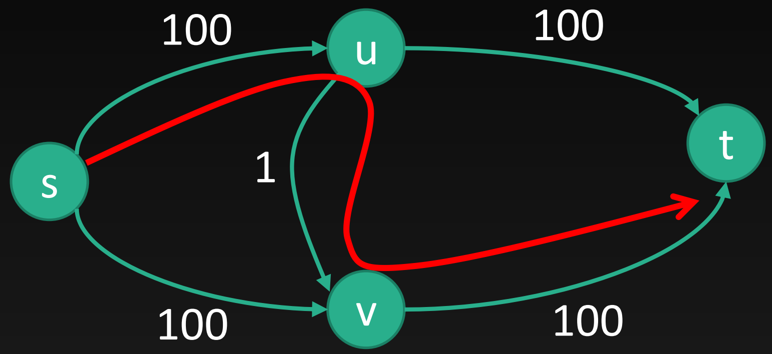
(v,u) is on the shortest path from s to u in G_i

$level_i(v) = level_i(u) - 1 \leq k - 2$

If there is no path from s to v then $level_{i+1}(v) = \infty$ and lemma is also true for v.

Bottleneck edge

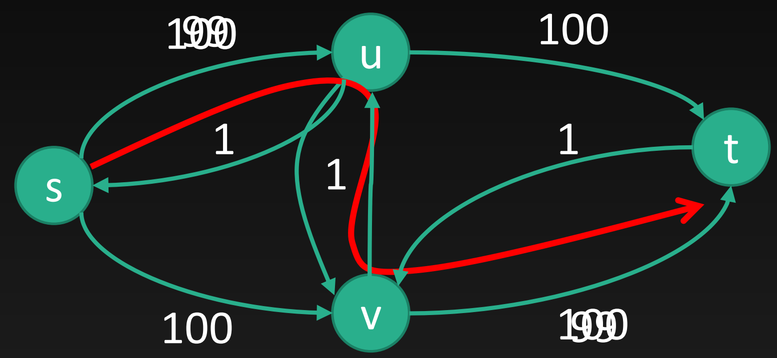
G_f



Edge e is bottleneck if residual capacity of e is minimum among edges on augmenting path

Bottleneck edge disappears after augmentation

G_f



How many times can $u \to v$ be bottleneck?

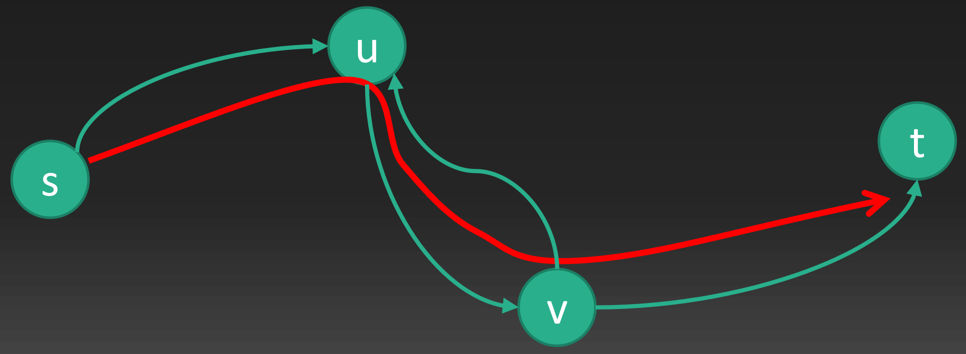
Lemma. Edge $u \to v$ can be bottleneck at most V/2 times.

Proof. Suppose $u \to v$ is bottleck for i^{th} augmenting path.

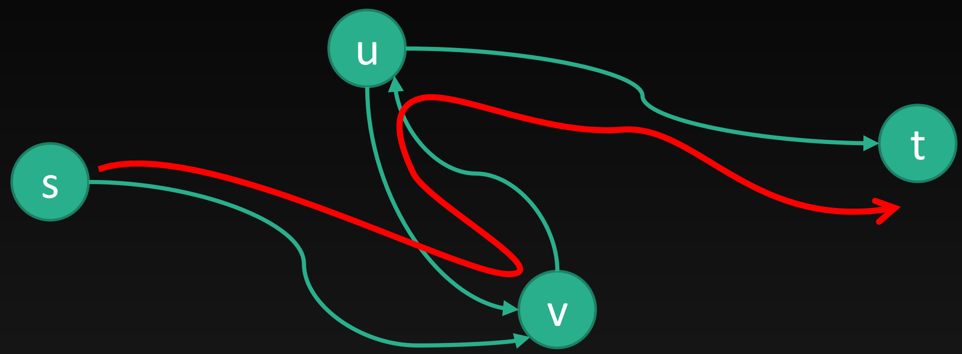
$u \to v$ is on shortest path in G_i so $level_i(u) + 1 = level_i(v)$.

$u \to v$ disappears in residual graph afterwards.

For $u \to v$ to be bottleneck again it must be reintroduced later.



How many times can $u \to v$ be bottleneck?



$u \to v$ reappears after j^th augmentation only if $v \to u$ is on j^th aug. path.

$v \to u$ is on shortest path in G_j so $level_j(u) = level_j(v) + 1$.

But we have $level_j(v) + 1 \geq level_i(v) + 1 = level_i(u) + 2$.

Thus, level(u) increases by at least 2 before $u \to v$ can be bottleneck again.

level(u) increases up to V times throughout algorithm $(0, 1, \ldots, V-1, \infty)$.

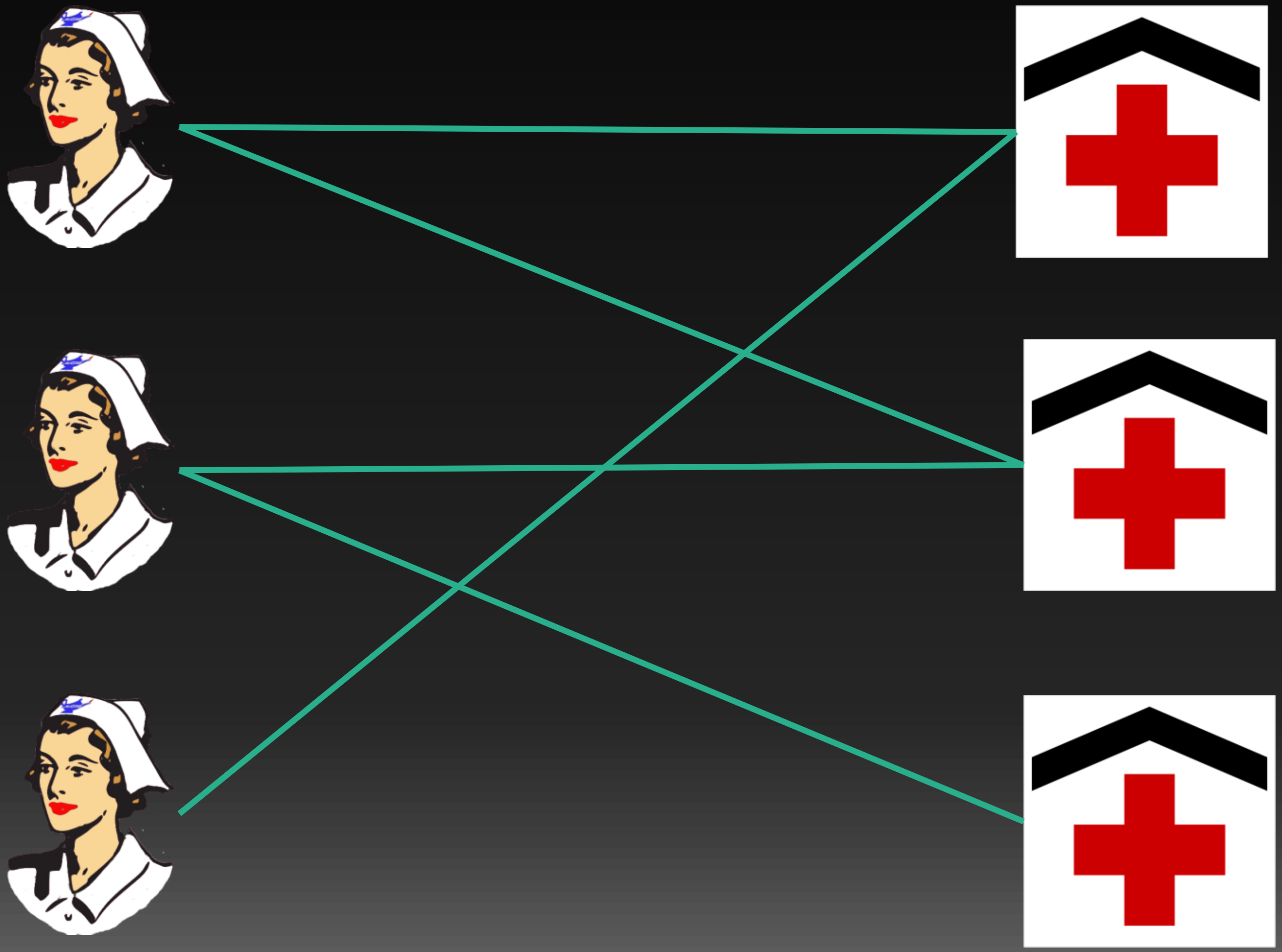
Thus, $u \to v$ can be bottleneck at most V/2 times.

Running time of Dinitz/Edmonds-Karp

- Each augmenting path has 1 bottleneck edge
- Each edge can be bottleneck V/2 times
- Thus, at most VE/2 augmentation steps
- Finding a path requires 1 BFS (O(V+E) time)
- Total running time O(VE(V+E))

Bipartite matching

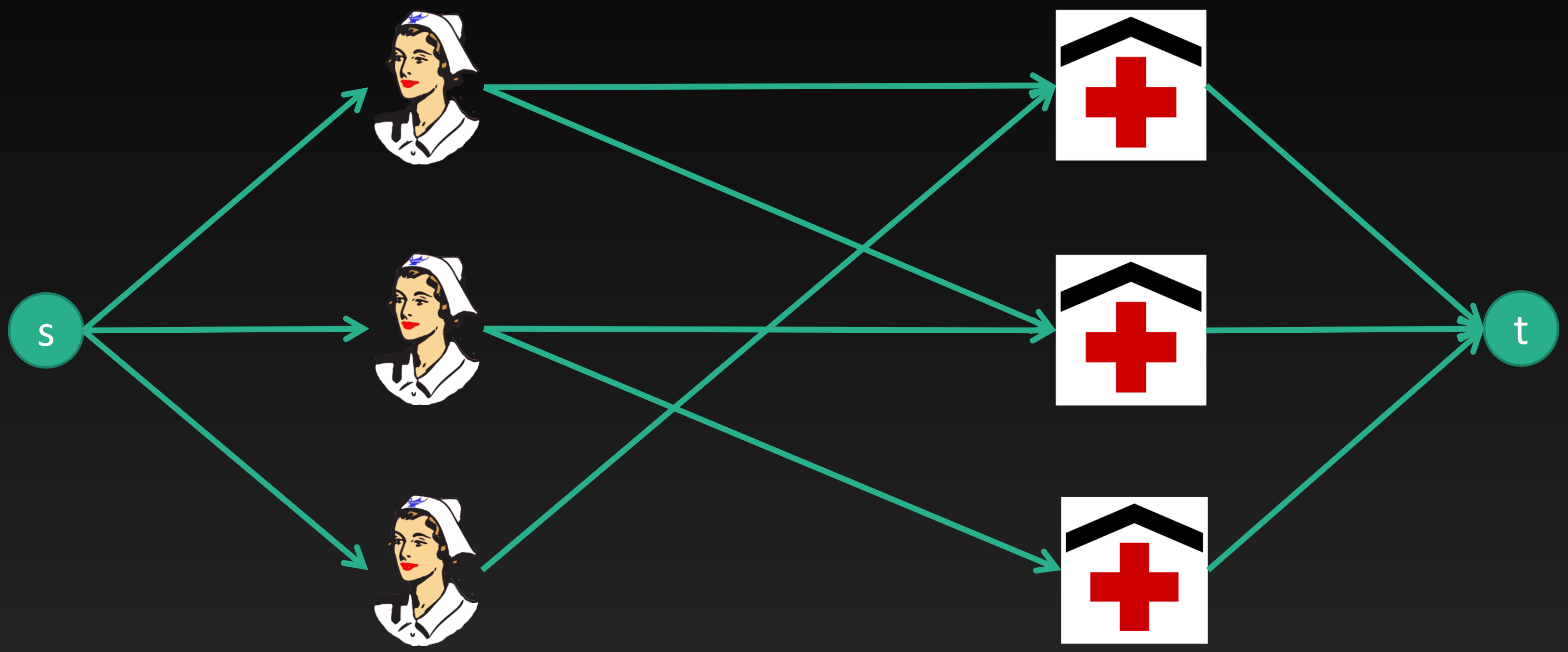
Bipartite matching



Bipartite matching

- Given graph $G = (L \cup R, E)$ where the edges are between L and R
- Find the largest subset $M \subseteq E$ such that each vertex is incident to at most one edge in M

Reduction to max flow



All edges have capacity 1

Find max flow and return all middle edges e with f(e)=1