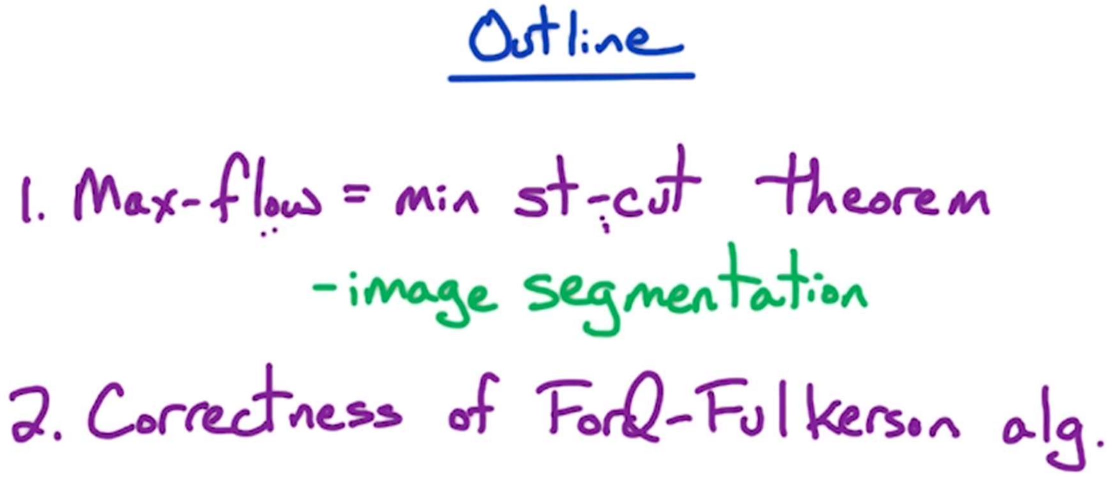
LESSON: MF2: Max-flow Min-cut

***

(Slide 1) Max-flow Min-cut image Lecture Outline



Outline

1. Max-flow = min st-cut theorem

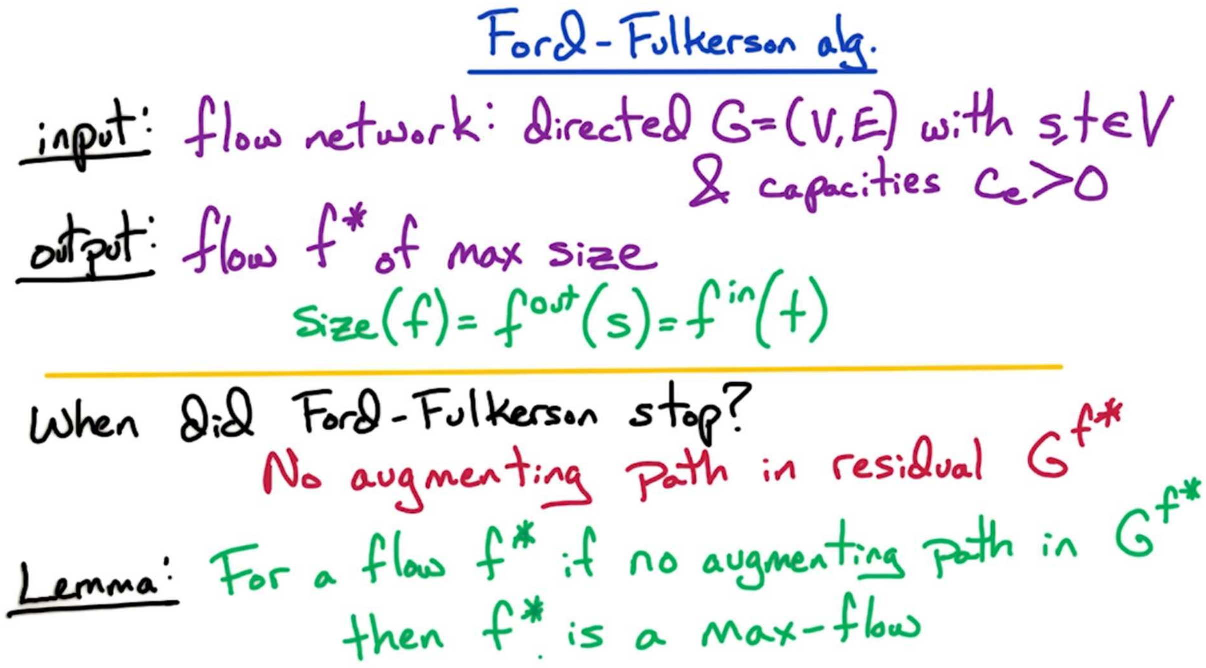
-image segmentation

2. Correctness of Ford-Fulkerson alg.

In this lecture, we're going to have two main tasks. The first task, is to prove the Max-flow Min-cut theorem. Later, we're going to use this theorem to solve the Image Segmentation problem. Our second main task, is to prove correctness of the Ford-Fulkerson algorithm. And, in fact, we will prove correctness of the Edmonds-Karp algorithm at the same time. Now this is going to follow as a byproduct of the Max-flow Min-cut theorem. Let's start with a quick overview of what we've seen so far for the Ford Fulkerson algorithm.

***

(Slide 2) Recap: Verifying Max-Flow



Ford-Fulkerson alg.

input: flow network: directed $G=(V,E)$ with $s,t \in V$ & capacities $c_e > 0$

output: flow f^* of max size

$\text{size}(f) = f^{out}(s) = f^{in}(t)$

When did Ford-Fulkerson stop?

No augmenting path in residual G^{f^*}

Lemma: For a flow f^* if no augmenting path in G^{f^*} then f^* is a max-flow

The input to the Ford–Fulkerson algorithm, or to the Max-flow problem in general, is a flow network. This is a directed graph G with a specified start and end vertex - s and t. And for each edge of the graph, we've specified a positive capacities c_e.

Now, if you recall for the Ford–Fulkerson algorithm, to analyze the running time, we required that the capacities were integer value. Let's ignore that assumption for now and let's look at the more general Max-flow problem. The output for the algorithm or for the Max-flow problem is a flow (f*) of maximum size. And recall that the size of the flow is the flow out of the source vertex or the flow into the sync vertex T.

Now recall, we analyzed the running time of the Ford–Fulkerson algorithm but we didn't prove that the flow that's outputted by the Ford–Fulkerson algorithm is of maximum size. So let's look at that now.

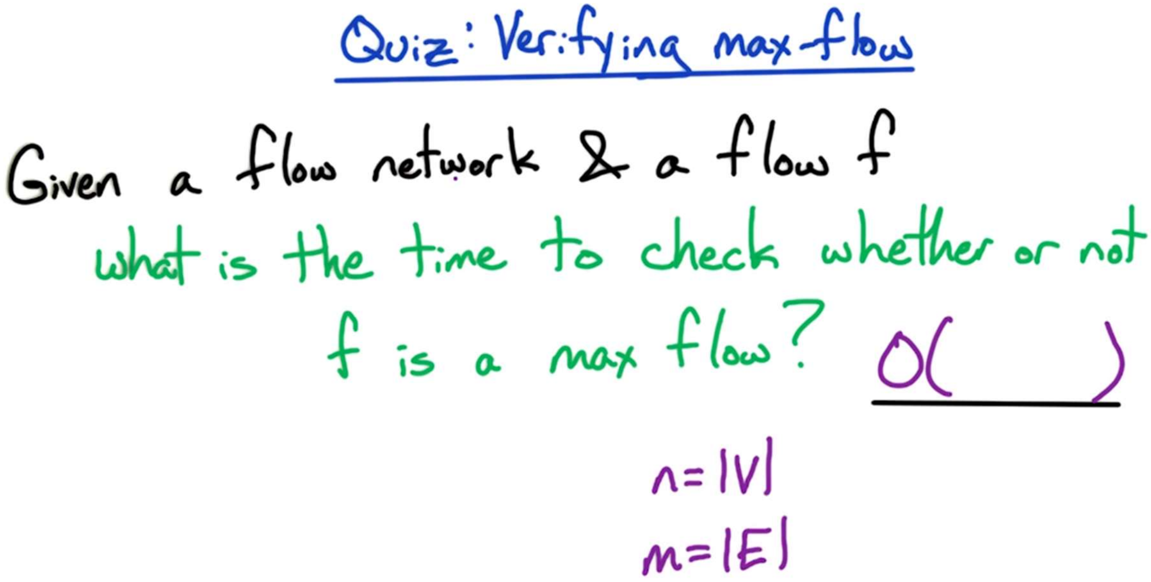
What do we know about this flow that's outputted by the Ford–Fulkerson algorithm? In particular, when did the Ford–Fulkerson stop? What is the stopping condition for the algorithm? - the algorithm stops when we look at the residual graph for the current flow.

So this is G^{f*}. Now, in this residual graph, we look for a path from s -> t. Such a path is called an augmenting path. If there's no augmenting path, there's no path from s -> t in this residual graph, then the algorithm stops. We have no way to augment the flow to get a bigger flow.

We're going to prove the following general lemma: for a flow f*r, if there's no augmenting path in the residual graph G^{f*}, then its flow is a max flow. So, if in the residual graph there's no path from s -> t, then this flow is a max flow.

***

(1st Slide 3) Quiz: Verifying Max-Flow



Quiz: Verifying max-flow

Given a flow network & a flow f

what is the time to check whether or not f is a max flow? O(   )

$n=|V|$

$m=|E|$

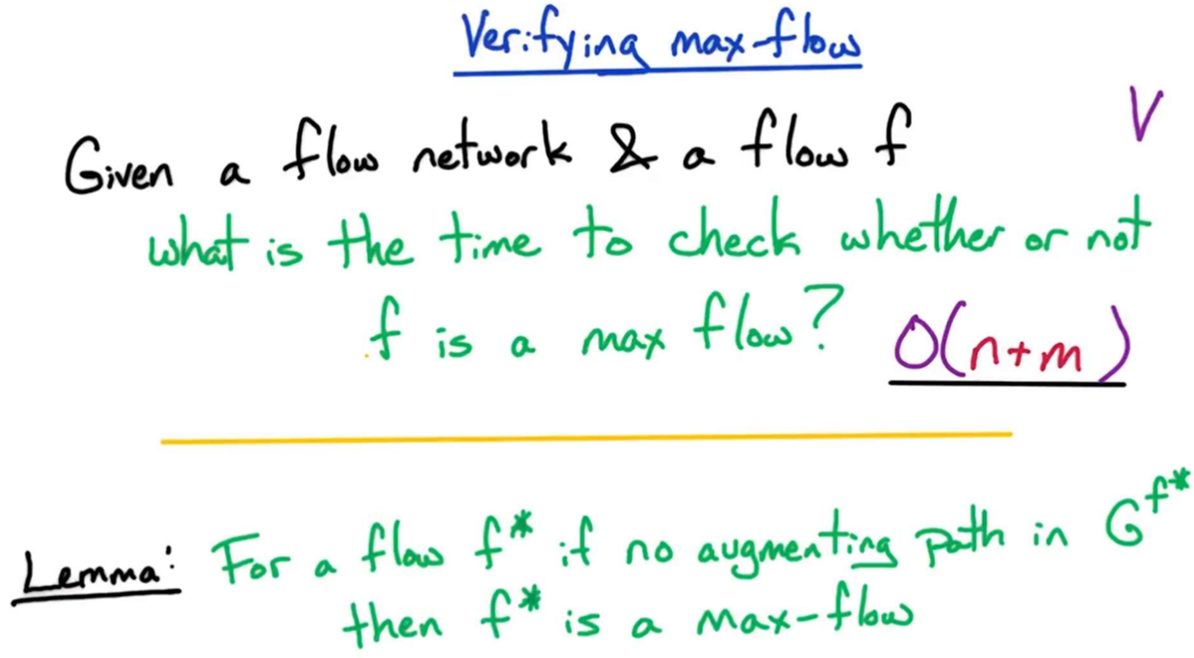
Let's look at an important consequence of dilemma we just stated on the previous slide. Given a flow network, so an input to the Max-flow problem, and given a particular flow f, what is the time required to check whether this particular flow f is a max flow or not? Specify your answer in big O notation, in terms of n, the number of vertices in the flow network, and m, the number of edges in the flow network.

MF2: Max-flow Min-cut: Quiz: Verifying Max-Flow

Question: Given a flow network and a flow f, what is the time to check whether or not f is a max flow? n = |V|, the number of vertices in the flow network and m = |E|, the number of edges in the flow network.

***

(2nd Slide 3) Quiz: Verifying Max-Flow (Answer)



Verifying max-flow

Given a flow network & a flow f ✓

what is the time to check whether or not f is a max flow? $O(n+m)$

Lemma: For a flow f^* if no augmenting path in G^{f^*} then f^* is a max-flow

The solution is linear time. It takes O(n+m) time to check whether a particular flow f is maximum or not.

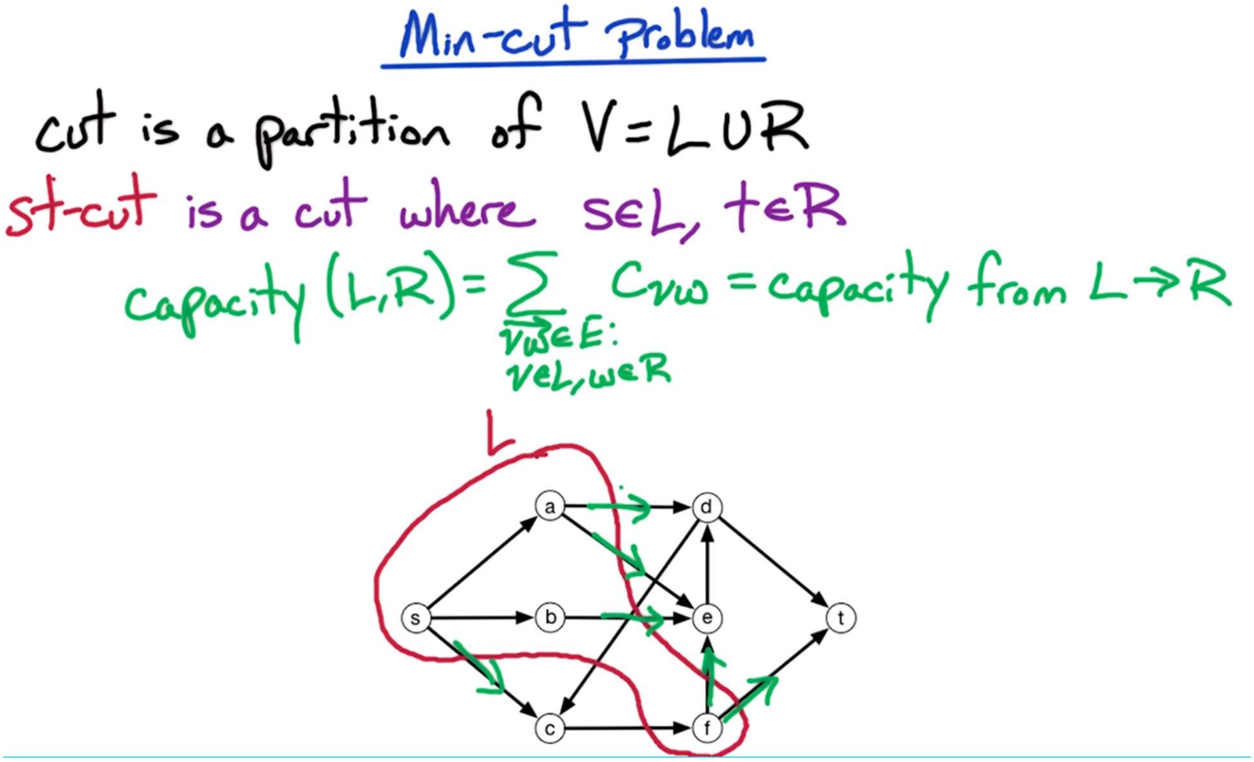
This follows as a consequence of the lemma we stated on the previous slide. For a flow f*, if there's no augmenting path, no path from s -> t in the residual graph for this particular flow, then this flow f* is a max-flow.

So to check whether f is a max-flow, we build its residual graph which takes linear time, O(n+m) time, and then we check for a path from s -> t in this residual graph. To do that, we simply run DFS from s to check whether t is reachable from s. What this lemma says is if there is no path from s -> t in this residual graph, then this particular flow f is a max-flow. So we output Yes.

Now what if there is a path from s -> t? Well, then we can augment a flow along this path. And then we get a flow of larger value. And therefore, this particular flow f is not a max flow because we can build a flow of larger size. Now, we'll get back to the proof of this lemma shortly. But let's go back first to the Max-flow Min-cut theorem, and let's see what Min-cut actually means.

***

(Slide 4) Min-cut Problem



Min-cut Problem

cut is a partition of $V = L \cup R$

st-cut is a cut where $s \in L$, $t \in R$

$$\text{capacity}(L,R) = \sum_{\substack{\vec{vw} \in E: \\ v \in L, w \in R}} c_{vw} = \text{capacity from } L \rightarrow R$$

Now a cut of the graph is a partition of the vertices into two sets.

We will think of these two sets as the left side and the right side. So we denote it as L and R. Then we are going to be looking at st-cuts. These are a particular type of cuts. This is the cut which separates s from t. So s is on one side - let's say the left side - and t is on the right side.

Here's our example of a flow network that we looked at earlier. Let's mark an example of an st-cut in this graph … here's an example of an st-cut … we put these four vertices s, a, b and f in the set L. Notice this set doesn't have to be a connected set. f is not connected to a and b in this subset. In the set R are the remaining four vertices, c, d, e and t.

Now we're going to be interested in the capacity of this st-cut. What exactly is the capacity? - recall, it's a directed graph. We're going to look at these directed edges v => w where v is in the left side and w was in the right side. So, these are all edges which go from the set L to the set R and we're going to look at the capacity of these edges.

So we're looking at the total capacity of edges from the set L -> R. But we're not looking at edges from R -> L. So we're looking at sum of the capacities of these six edges which exit L. Notice that these edges such as c -> F which go into L do not count in this count.

***

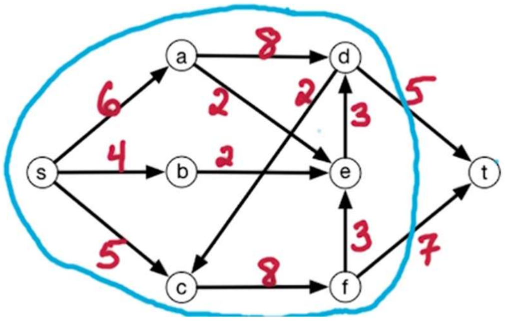
(Slide 5) Problem Formulation

Problem formulation

Min st-cut problem:

input: Flow network

output: st-cut (L,R) with min capacity



Let's formally state the Min st-cut problem. The input is a flow network. So, we have a directed graph with a specified start and end vertex. And, for every edge, we have a positive capacity.

Now the output for the Min st-cut problem is an st-cut (L, R). So, so s has to be an L and t has to be an R. And, we want to find that st-cut with minimum capacity.

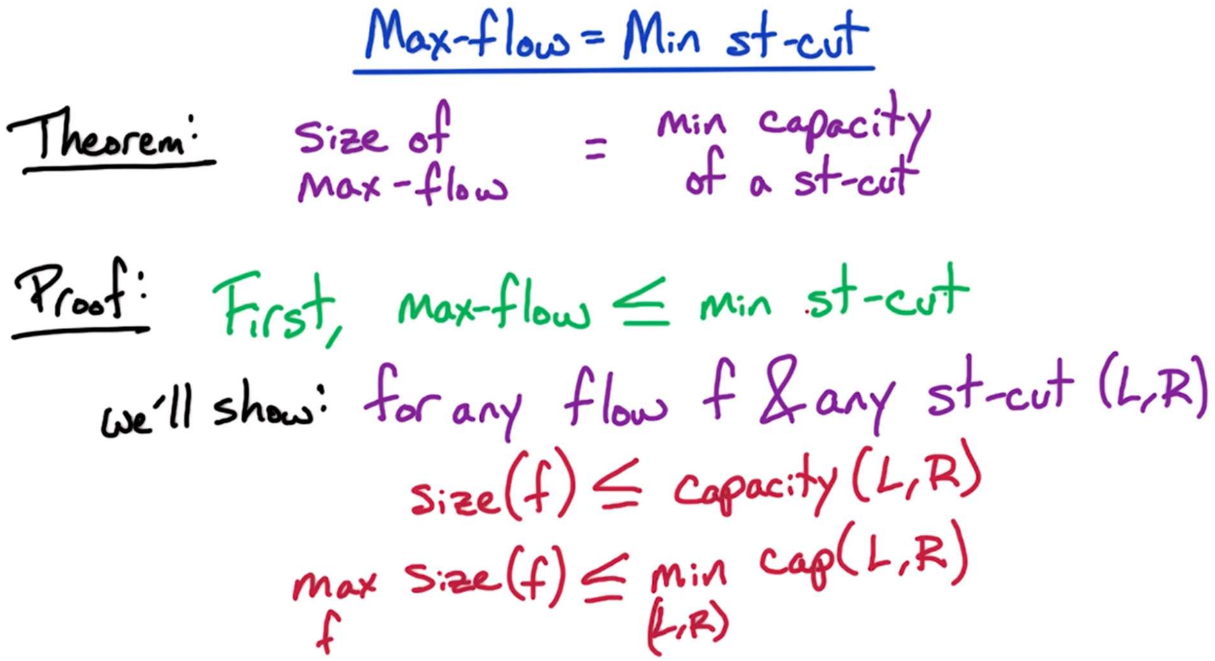
Recall the capacity of this cut is the sum of the capacities of the edges that go from L to R, and we ignore the edges that go from R to L. Now let's take a look at this cut that we looked at earlier.

So the set L contains s, a, b and f. The capacity of this cut is 8 + 2 + 2 + 5 + 7 + 3. So the capacity is 27. We want to find the cut of minimum capacity.

Here is an st-cut of minimum capacity. R simply contains t and L contains everybody else. The capacity of this cut is 12. We just have these two edges exiting L -> R. How do we know this is min st-cut? - well, we're going to prove the Max-flow Min-cut theorem which says that the size of the max flow equals the size of the min st-cut. We know there's a flow in this graph of size 12; therefore, we know that this flow and this st-cut are both optimal.

***

(Slide 6) Max-Flow = Min st-Cut



Max-flow = Min st-cut

Theorem: Size of Max-flow = min capacity of a st-cut

Proof: First, max-flow ≤ min st-cut

we'll show: for any flow f & any st-cut (L,R)

$$\text{size}(f) \leq \text{capacity}(L,R)$$

$$\max_{f} \text{size}(f) \leq \min_{(L,R)} \text{cap}(L,R)$$

The Max-flow Min-cut theorem states that for any flow network, the size of the max-flow equals the minimum capacity of an st-cut.

Just a quick note about notation here:

- We keep calling this an st-cut to distinguish that s is on one side of the cut and t is on the other side of the cut. Notice s and t are also implicit in this max-flow formulation.

- We could have called it a max st-flow because we're sending the flow from s -> t. Now in the max-flow problem, it's clear there must be a start vertex s and an end vertex t. In a Min-cut problem, there doesn't necessarily have to be an s and t that you're separating. We could instead look at the Min-cut overall without looking at whether we separate s and t.

  So that's why we include the s and t in this formulation. So we called it the Min st-cut problem to differentiate it from the Min-cut problem.

Now, let's dive into the proof of this Min-cut Max-flow theorem. First, we're going to prove that the size of the max flow is the most the minimum capacity of an st-cut. And then, we'll prove the reverse inequality.

So we'll prove that the max-flow is at least the minimum capacity of an st-cut. Once we prove both inequalities, then we've proved equality, and then we're done. To prove this inequality, what we'll show is that for any flow and any st-cut (L, R), the size of this flow is at most the capacity of this cut.

Notice that this simpler result will imply this inequality. Why is that the case? - well, this is true for any flow f, so we might as well take a max flow and it's still true. So if we maximize over f, the size of f is still at most the capacity of this cut. And since the right hand side holds for any st-cut, we might as well take the cut of smallest size. So, for the right hand side we can choose any cut we want. So, we'll choose the cut of minimum capacity, and then we have that the max-flow size is the most the min st-cut capacity.

So that shows the max-flow is the most of min st-cut. So we proved this inequality. So we have a much simpler task now. We can forget about the max flow and the min cut. We simply have to take any flow f, and any st-cut and show that the size of this flow is at most the capacity of this cut. And we don't have to worry anything about showing that this flow is optimal or this cut is optimal.

***

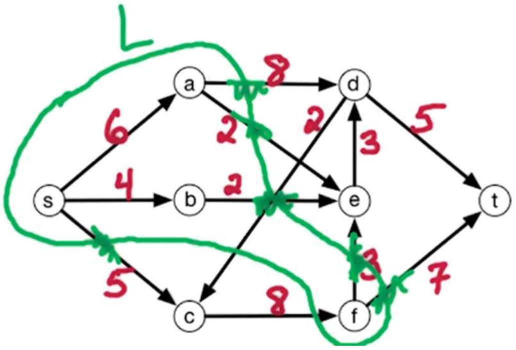
(Slide 7) Max-Flow Min st-Cut

Max-flow $\leq$ min st-cut

we'll show: for any flow f & any st-cut (L,R)

$\text{size}(f) \leq \text{capacity}(L,R)$

Claim: $\text{size}(f) = f^{out}(L) - f^{in}(L)$



Recall, we want to show that the size of the max-flow is at most to the capacity of the min st-cut. And to do that, we need to show that for any particular flow f and any particular st-cut (L,R), the size of this flow is at most the capacity of this cut.

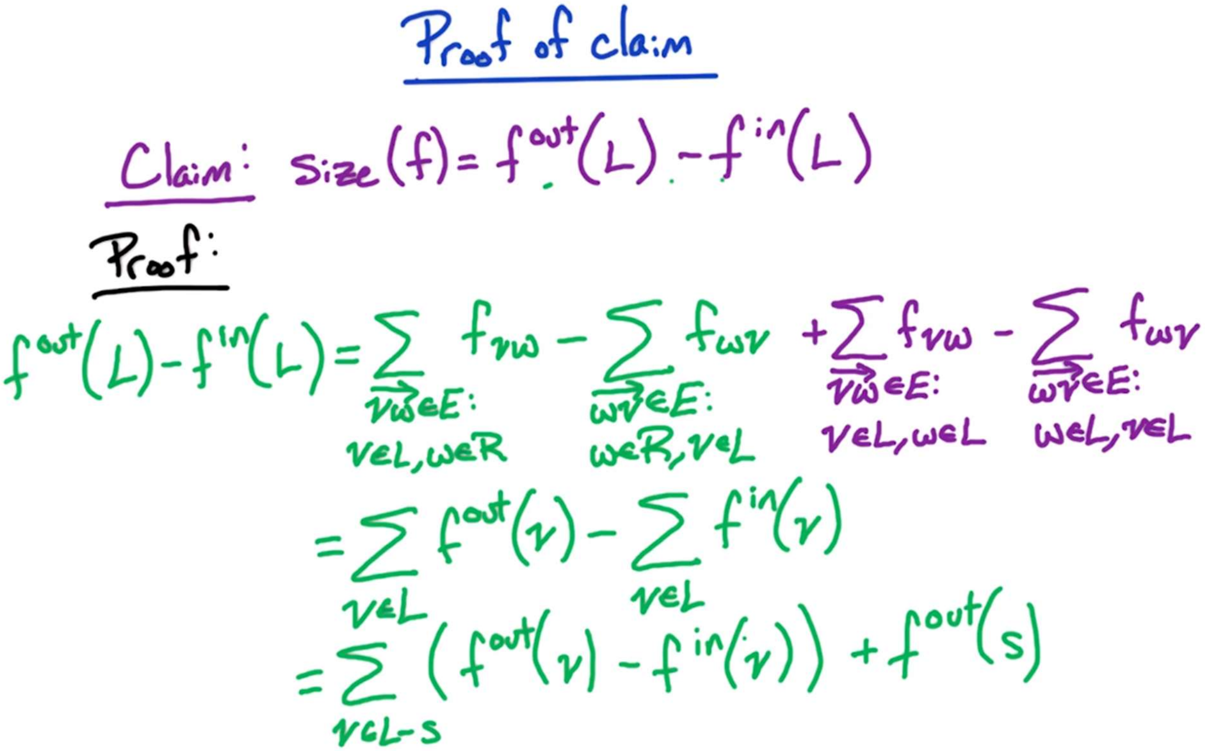
Let's take a look again at our earlier example of a flow network. And, here is our example of an st-cut. These four vertices are set L and the other four vertices are the set R. Now, the capacity of this cut is 27. That's the total capacity of the edges going from L to R.

Now the intuition for this claim is that this flow is going from s -> t. At some point, it has to exit from L to R. These six edges which go from L to R are the only edges that the flow can escape L from. Now the total capacity of these edges is 27. So the total amount of flow that can go along these edges is at most 27. So, the size of this flow has to be at most 27. That's the intuition. Let's formalize it.

To formalize the intuition we just stated, we're going to prove the following claim: We're going to prove that the size of this flow equals the total flow out of the set L minus the total flow into the set L. Now it will just be a few lines to prove this claim. It's quite easy. And then once we have this claim, the main statement that the size of f is at most of capacity of this cut will follow immediately. And this claim is going to be useful later. We're going to use it when we prove the reverse inequality, that the max flow is at least the min st- cut capacity.

***

(Slide 8) Proof of Claim (Max-flow Min-cut)



Proof of claim

Claim: $\text{size}(f) = f^{out}(L) - f^{in}(L)$

Proof:

$$f^{out}(L)-f^{in}(L)=\sum_{\substack{\vec{vw}\in E:\\ v\in L, w\in R}} f_{vw} - \sum_{\substack{\vec{wv}\in E:\\ w\in R, v\in L}} f_{wv} + \sum_{\substack{\vec{vw}\in E:\\ v\in L, w\in L}} f_{vw} - \sum_{\substack{\vec{wv}\in E:\\ w\in L, v\in L}} f_{wv}$$

$$=\sum_{v\in L} f^{out}(v) - \sum_{v\in L} f^{in}(v)$$

$$=\sum_{v\in L-s}\left(f^{out}(v)-f^{in}(v)\right) + f^{out}(s)$$

Let's prove this claim that for any flow f and for any st-cut (L, R), the size of this flow equals the total flow out of the set L minus the total flow into the set L.

Claim: $\text{size}(f^*) = f^{out}(L) - f^{in}(L)$

Let's look at the right hand side, so we have the flow out of the set L minus the flow into the set L. Now, the first term - the flow out of the set L - that's the sum of edges of the graph, where the head of the edge is in the set L, and the tail of the edge is outside of the set L. And we look at the flow along that edge - that's the flow out of the set L.

Now, for the flow into the set L, we're going to sum over edges w -> v, where w is outside the set L, and v is inside the set L, and we look at the flow along that edge w -> v.

Now I've labeled these edges vw, and this one wv, so that v is always the vertex in the set L. Now, this captures the flow out of L, and the flow into L. Now - want to add an additional term - so, we're going to add it and subtract it that at the same time, and therefore we still have equality.

I'm going to look those edges vw, where $v \in L$, and $w \in L$ and I'm going to add in the flow along those edges, and I'm going to subtract off the same term but I'm going to label the edge by wv instead of vw. But I'm going to be summing up over the same set of edges. Just labeled differently.

Now, take the first and the third term - these are the positive terms. This considers the edges v -> R, and this considers the edges v -> L. So, if I combine them together, I have all edges out of v. This first term and this third term combined give us the sum of all vertices in the set L of the total flow out of this vertex L. Similarly, this term and this term give the flow into the vertex v because this term gives the flow of the edges from R -> v, and this gives from L -> v. So, combined they give the total flow into v.

Now let's distinguish one particular vertex. The set L contains the source vertex, s. What do we know about the source vertex? Well, the flow in is zero. So we only have the flow out of it. So, let's take out s from this set L. So now we have a sum of vertices V in the set L – {s}, and we have the flow out of this vertex v minus the flow into this vertex v.

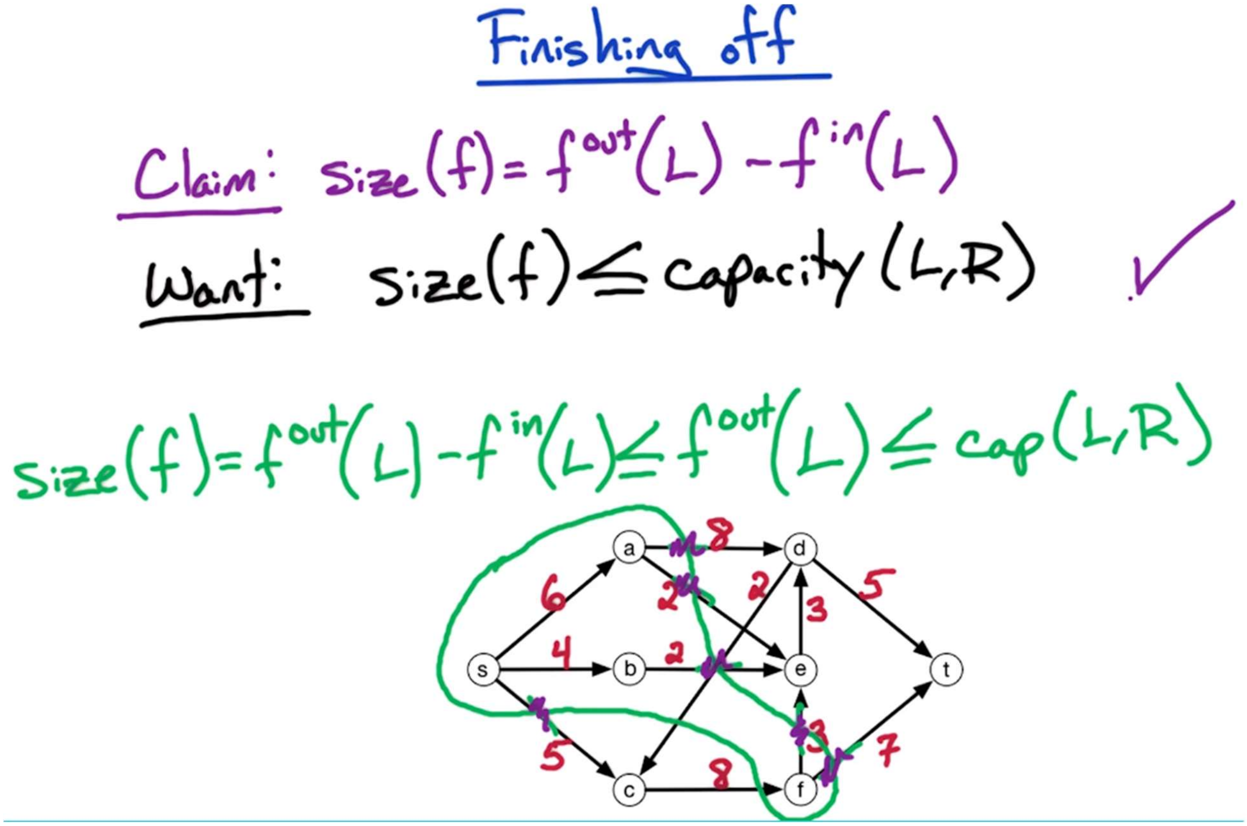
We dropped the term for s, so we have the flow out of s. We also dropped the term for the flow into s but we said that's zero, so we can drop it.

Now, what do you notice for all these vertices which are not s? The flow out has to equal the flow in. That's the definition of a valid flow, conservation of flow. So these two terms are equal, so this whole sum is zero, and all we're left with is the total flow out of s which is the size of f.   The total size of this flow is equal to the flow out of s.

Therefore, we've shown that the size of f equals the flow out of L minus the flow into L which is the claim that we wanted to prove.

***

(Slide 9) Finishing Off



Finishing off

Claim: $\text{size}(f) = f^{out}(L) - f^{in}(L)$

Want: $\text{size}(f) \leq \text{capacity}(L,R)$ ✓

$$\text{size}(f) = f^{out}(L) - f^{in}(L) \leq f^{out}(L) \leq \text{cap}(L,R)$$

So now we've proven this claim, the size of f equals the flow out of L minus the flow in it to L, and this is true for any flow f, and for any st-cut (L,R).

Now, recall we want to show - that the size of this flow f is at most the capacity of this st-cut (L, R). And, we saw before that this inequality implies that the max flow is at most the min st-cut capacity. So, let's see how this claim implies this desired result.

Well, the size of F equals the flow out of L minus the flow into L - that's just the claim again. Where we can drop this second term, and it only gets bigger. So, this is at most the flow out of L.

Now, along every edge that leaves the set L - the flow along that edge has to be at most the capacity of that edge. Therefore, the total flow out of this set L, has to be at most the capacity.

Because the capacity is defined as the total capacity from the set L -> R, we didn't count the edges from R -> L.

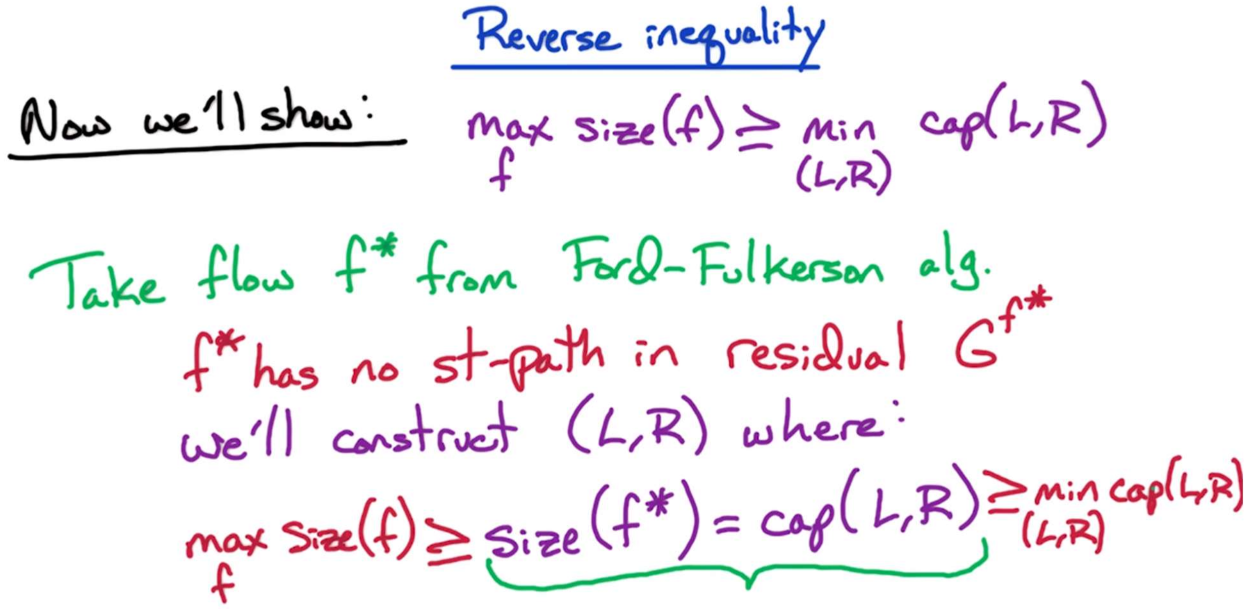
Looking back at our earlier example, for this st-cut (L,R), the capacity of this cut is 27, and the total flow along these six edges, which leave this set L, clearly, must be at most the total capacity along these edges, which is 27. So, the flow out of this set L has to be at most the capacity of this cut. So, this proves that desired result.

And therefore, we have that the max flow size is at most the min st-cut capacity.

Now, to complete the proof of the theorem, we have to prove the reverse inequality - that the max flow size is at least the min st-cut capacity. Let's dive into that now.

***

(Slide 10) Reverse inequality



Reverse inequality

Now we'll show: $\max_{f} \text{size}(f) \geq \min_{(L,R)} \text{cap}(L,R)$

Take flow f^* from Ford-Fulkerson alg.

f^* has no st-path in residual G^{f^*}

we'll construct (L,R) where:

$$\max_{f} \text{Size}(f) \geq \underbrace{\text{Size}(f^*) = \text{cap}(L,R)}_{} \geq \min_{(L,R)} \text{cap}(L,R)$$

Now, we'll show the following inequality: we'll show that max over the flows, f, the size of f, that the max flow size is at least the capacity of the min st-cut.

We just proved that the max-flow size is at most the min st-cut capacity. Now if we prove this inequality, then we shown that its equality - that the left hand side equals the right hand side.

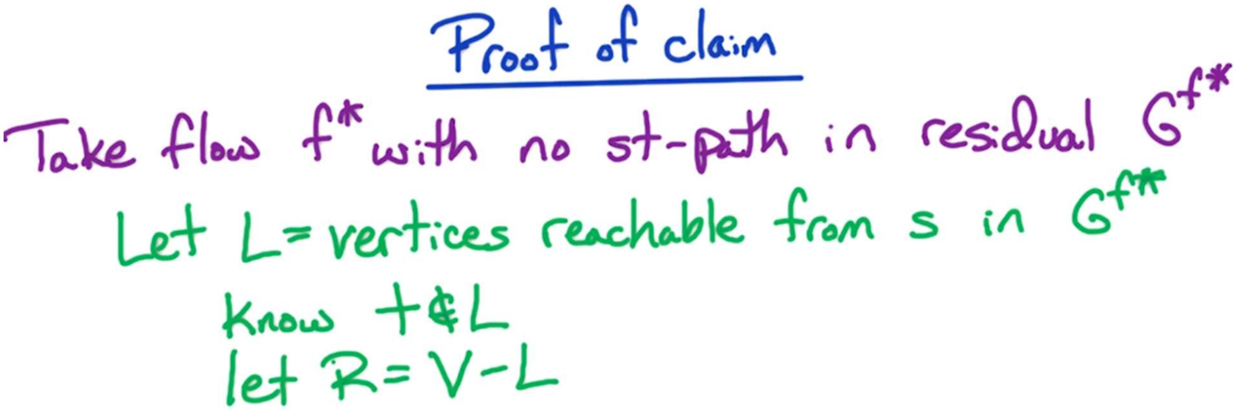
To prove this inequality, what we're going to do is we're going to take the flow f*, produced by the Ford-Fulkerson algorithm. What do we know about this flow? The Ford-Fulkerson algorithm stops when there's no augmenting path in the residual network. So we know that this flow, f*, has no path from s -> t in this residual network for this flow. What we're going to prove, is that for any flow, f*, which has no augmenting path in the residual network, we're going to construct an st-cut, (L, R) which has the following property: the size of this flow, f*, equals the capacity of this cut that we construct. Now once we prove this, we claim that it's done.

Why does this statement imply this inequality? = well, this is one particular flow, f*. How does this compare to the max flow size? Certainly, if we maximize over the flows f, the size of this flow, is at least the size f*. It maybe equality if f* is a max flow, but certainly, the max flow is at least the size of f*.

And how does the capacity of this particular cut (L, R) compare to the capacity of the min st-cut? Well certainly, the capacity of this particular cut is at least the minimum capacity of any cut. So if we prove this middle equality, that the size of F* equals the capacity of this cut (L,R) that we construct, then that implies that the max flow size is at least, the min st-cut capacity, which is what we want to prove. So it just remains to prove this equality.

***

(Slide 11) Proof of Claim (Max-flow Min-cut 2)



Proof of claim

Take flow f^* with no st-path in residual G^{f^*}

Let L = vertices reachable from s in G^{f^*}

know $t \notin L$

let $R = V - L$

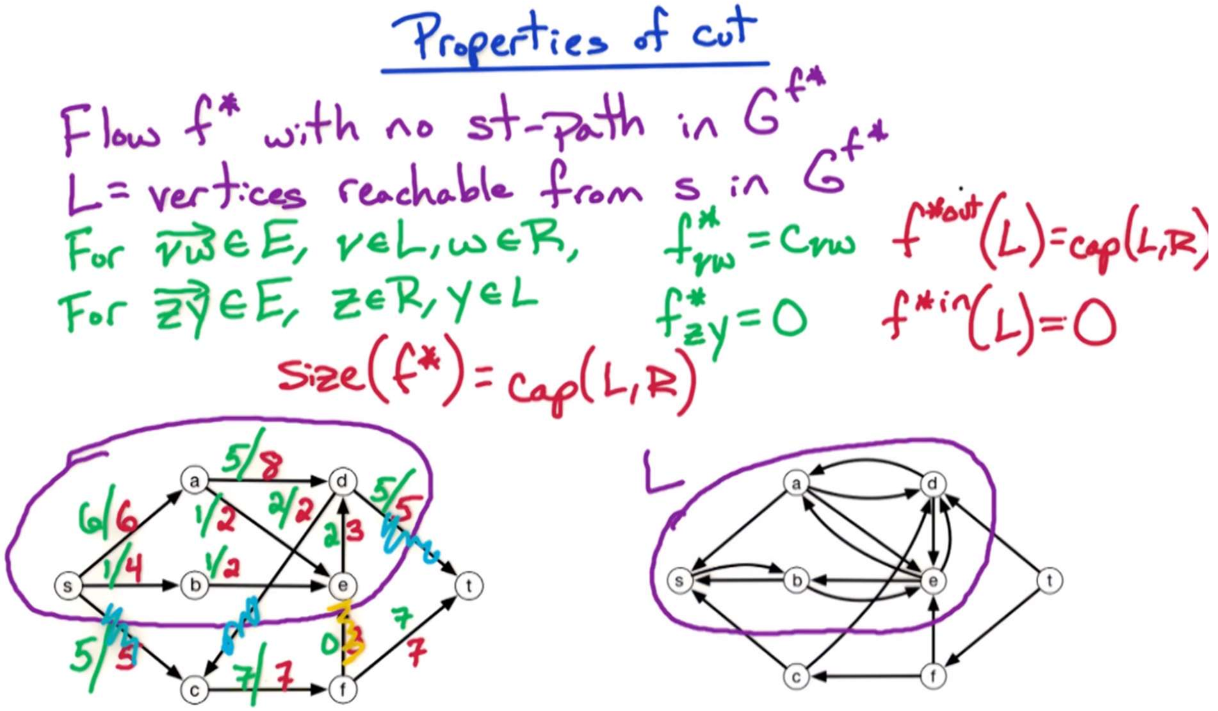
So let's go ahead and prove the claim. So let's take a flow f* that has no augmenting path in this residual network and we wanted to find an st-cut now.

We want to use this property about this f*. We know there's no st-path in this residual network. So, we know that t is not reachable from this vertex s in this graph. So that's a way to separate s from t. Therefore, we're going to let L be those vertices which are reachable from the vertex s in this residual network. Now we know this vertex t is not in this set L, why? Because we know there's no path from s -> t in this residual network - that's our assumption about this flow f*. Therefore, we let R be the remaining vertices and we know that the vertex t is in this set R. So s is clearly reachable from itself. So s is in L and t is in R. So this set (L,R) is an st-cut.

So we've constructed our st-cut, now we have to prove that this st-cut has capacity which is equal to the size of this flow.

***

(Slide 12) Properties of Cut



Properties of cut

Flow f^* with no st-path in G^{f^*}

$L=$ vertices reachable from s in G^{f^*}

For $\overrightarrow{vw} \in E$, $v \in L, w \in R$, $f^*_{vw} = c_{vw}$ $f^{*out}(L) = cap(L,R)$

For $\overrightarrow{zy} \in E$, $z \in R, y \in L$ $f^*_{zy} = 0$ $f^{*in}(L) = 0$

$size(f^*) = cap(L,R)$

Let's take a look now at the properties of this cut we just constructed and let's use as an illustration our earlier example of a flow network.

Here's our example of a flow network G with the capacities marked in red. We're looking at a flow f* which has no st-path in the residual network G^{f*}.

Here's an example of such a flow f* with the green numbers indicating the flow. Now let's look at this residual network for this flow f*.

Recall some properties of this residual network. For example, this edge from e -> d has positive flow so the reverse edge from d -> e appears in this graph. Also, the edge from e -> d has some spare capacity available so the edge from e -> d itself appears there.

Similarly, this edge from d -> c has some flow along it so the reverse edge from c -> d appears. But the edge from d => c has no leftover capacity available so the forward edge does not appear in this residual network.

Now the capacities on these edges in the residual network are not important. What's important are simply which edges are present or not. We define this set L to be the set of vertices reachable from the source vertex in this residual network. Let's look at the set L in this example.

So certainly s itself is reachable from s; also, b is reachable and then we reach e and d and a. So these five vertices are the set L and the remaining three vertices are the set R. And, that defines our st-cut (L,R). By assumption, t is not reachable from s in this residual network therefore {t} is an R, so it's a valid st-cut.

Let's look at some properties of this st-cut LR. For convenience, let me draw this L on the original graph as well. Let's look at edges in the original graph which go from the set L -> R.

d -> t is one such example; d -> c is another example; and, s -> c is another example. Now, notice that none of these edges appear in the residual network. Why? - because L is the set of vertices reachable from s. So, if this edge from d -> t appeared in this residual network, then t would be added into the set L. It would be reachable from s and then we'd have a contradiction with the definition of L.

So, for any edge in the original graph, say from v -> w, if this edge leaves the set L, then $v \in L$ and $w \in R$. Well, we know that this edge does not appear in the residual network, so what does that say about the flow along this edge? - this edge must be fully capacitated and therefore the forward edge does not appear in the residual network; so, the flow along this edge equals its capacity.

Now since every edge from L -> R is fully capacitated, what does this say about the total flow out of L? Well it's equal to the capacity from L -> R. The total flow out is equal to the total capacity from L to R. Recall that this capacity was defined as edges going from L -> R. It doesn't count edges going from R -> L.

… well … that's good … we're close to what we want to prove.

Let's take a look at the edges that are going from R -> L in the original graph … there is only one such edge in this example, the edge going from f -> e. So, consider an edge z -> y in the original graph whereas $z \in R$ whereas $y \in L$.

What do we know about the flow along this edge (from z -> y), say from f -> e? - It's flow is zero, why? Because this edge from e -> f, the reverse edge does not appear in this residual network. If it did appear then f would be included in the set L which it's not. So, since the back edge does not appear in the residual network, then this forward edge has to have flow zero.

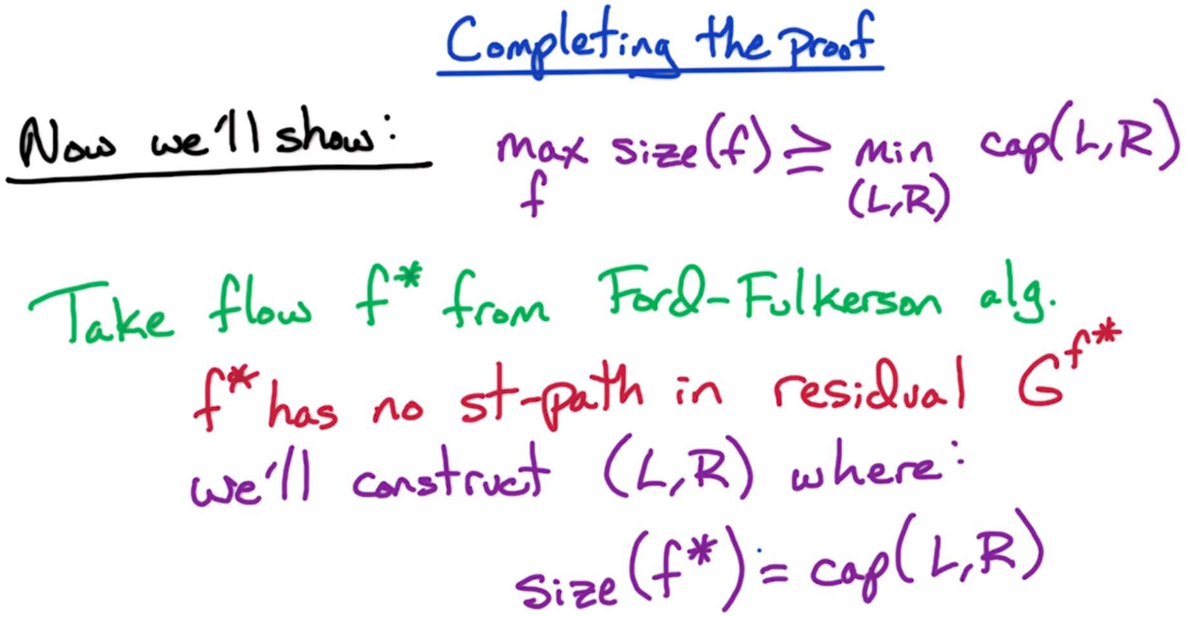
Since this is true for all edges into the set L, we conclude that the total flow into L is zero in this flow f*.

Now, we proved earlier for any flow f* and any st-cut, (L,R), that the size of this flow f* equals the flow out of L minus the flow into L. What we've seen so far for this st-cut (L,R), the flow out of this set L is equal to the capacity and the flow in equals zero. So, that means the size of this flow equals the capacity of this cut (L,R). This statement is exactly what we need to prove in order to finish the proof of the max flow main cut theorem.

So that completes the proof of the theorem.

***

(Slide 13) Completing the Proof



Completing the Proof

Now we'll show: $\max_{f} \text{size}(f) \geq \min_{(L,R)} \text{cap}(L,R)$

Take flow f^* from Ford-Fulkerson alg.

f^* has no st-path in residual G^{f^*}

we'll construct (L,R) where:

$\text{size}(f^*) = \text{cap}(L,R)$

Now let's go back to our earlier slide. We're trying to prove this inequality. We're trying to prove that the size of the max flow is at least the minimum capacity of an st-cut.

To do that, we took any flow which has no augmenting path in the residual network, and we constructed a st-cut. How do we construct the st-cut? We set L to be the set of vertices reachable from s, in this residual network. Since, there is no augmenting path, there is no st-path in this residual network. So, t is not reachable from s. So, t is in the set R and s is in the set L. So, this is a valid st-cut. And then, what we just proved is that the size of this flow, f*, is equal to the capacity of this cut. Therefore, the max flow is at least this size, and the min st-cut capacity is at most this. So, the max flow is at least the min st-cut capacity.

So we proved this inequality, and earlier we proved the other inequality, and therefore, that completes the proof of the max flow min cut theorem. We've proved that the max flow size equals the capacity of the min st-cut. Notice, we've also proven the earlier, lemma that we claimed.

We've shown, that for any flow that has no augmenting path in the residual network, we can construct an st-cut, where the size of this flow equals the capacity of the st-cut. Now the only way we can have equality here is if both of these are optimal. So this must be a max flow and this must be a min st-cut.

So, first off, this proves that Ford-Fulkerson algorithm, or any algorithm which stops when there is no augmenting path in the residual network, is correct. It outputs a max flow, but, secondly, it shows us a way of constructing a min st-cut.

If we take a max flow f*, and we set L to be those vertices reachable from s in the residual network, then that st-cut, that we just defined, has capacity equal to the size of this flow. So the capacity here must be optimal, this must be a min st-cut.

So we have a way of verifying that we have a max flow, and we have a way, given a max flow, of constructing an st-cut of minimum capacity, and were going to use that in our image segmentation application, and were going to use this lemma (about a flow being optimal if there is no augmenting path in the residual network) for proof of correctness of not only the Ford-Fulkerson algorithm, but also of the Edmonds-Karp algorithm.