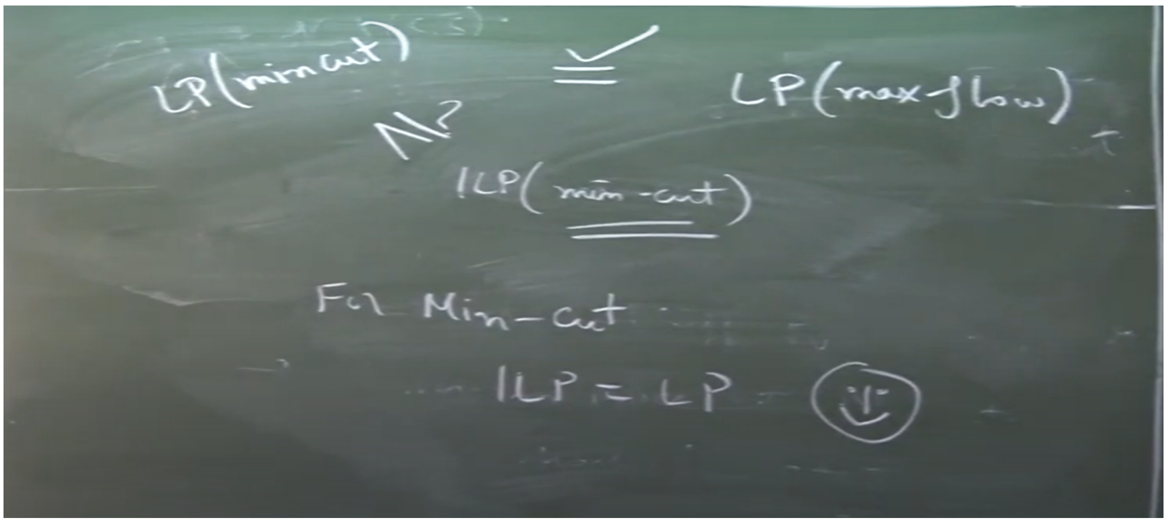
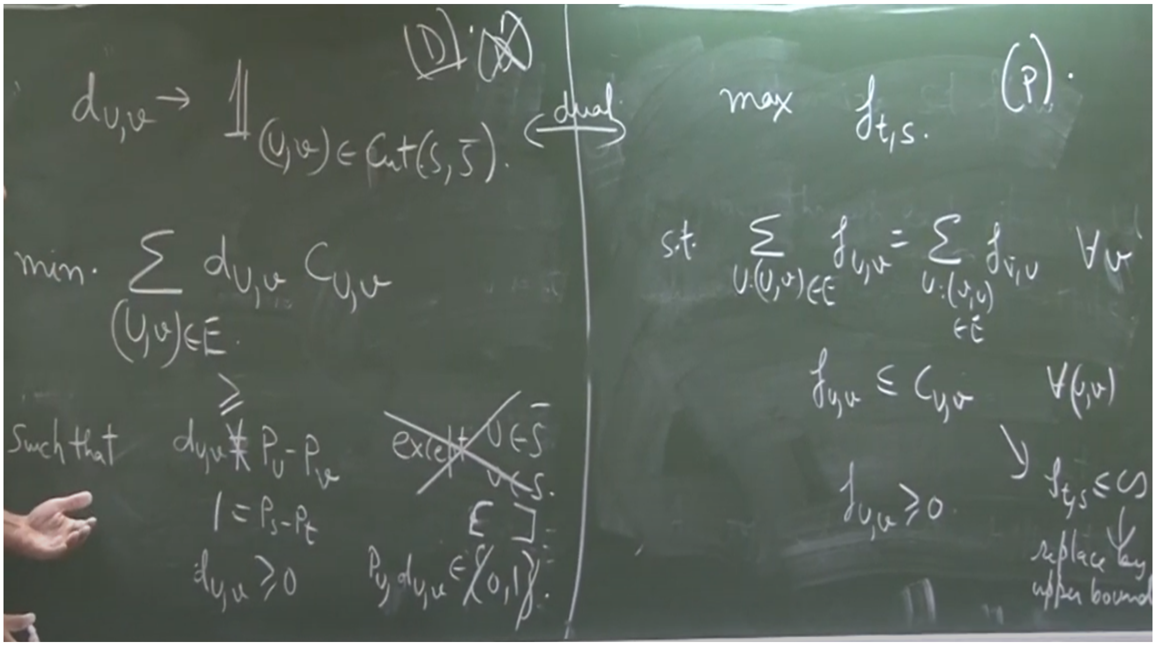
Linear Programming and its Applications to Computer Science
Prof. Rajat Mittal
Department of Computer Science and Engineering
Indian Institute Of Technology, Kanpur

Lecture – 41
Max Flow = Min Cut

So, now we have LP for max flow, we have ILP for min cut. And we have an LP for min cut, we know this, just because the feasible region has expanded. And if you want to relate these problems, because this is the actual solid min cut solution. If you want to relate these problems, we want to say how are these two related. And how approximately equal are these two quantities right. And what I am saying is that this ILP to LP conversion is very generic, we always ask that question.



But first let us look at the easy thing, the thing which is actually easy is to show that this is equal right. And this is your intuition, these programs are going to be dual of each other. Not a surprise now, almost most of you might have guessed it, I do not have time for it, I wanted to make you convert this program into this program as a dual. So, what I will do is probably sneak this into an end sem or a quiz.



So, make sure you know how to take this, convert this and get this exactly. This is another conversion, generic another conversion where lot of constraints create variables and you can already see right. When you look here right, there will be variables for vertices PUs, there will be variables for right. And you can simplify lot of things here right. For example, what can you say about duv, do I want to constrain it between 0 and 1 or I can say something more.

Do I need to enforce the constraint duv less than equal to 1, why? So, we already have enforced that duv is bigger than Pu minus Pv, it is a minimization problem. So, it will always be it will never be bigger than like bigger than Pu minus Pv right. So, that means if this is less than 1, this will automatically be less than 1. So, actually I can do not need to enforce this constraint, this will automatically be enforced. Now, with these things in mind you can actually check things work out very nicely.

There is just a single variable here fuv, there is only 1 constraint here. This is a constraint for another edge right. So, things match up you see that things match up right. Whatever variables I wanted same appear here, whatever constraints I wanted the numbers. Those are exactly the constraint, they are exactly the constraint which few with few simplifications.

So, that you should be able to make and if you do not learn it then you lose marks in either a quiz or a answer. So, make sure that you can take the dual of this. If you cannot post it on hello have a discussion, but make sure that this dual you can take. And this is generally easy part, ok. Looking at two problems people guess oh yeah these seems like dual of each other.

And then you write a linear program for one, write a linear program for another they match up. It might be hard for you, because you are not taking enough duals, but it is a easy part in general. So, this is fine. Most of the time the difficult part is this. How to relate these two quantities? We are going to relate this now and we are going to see this is actually easy in this case.

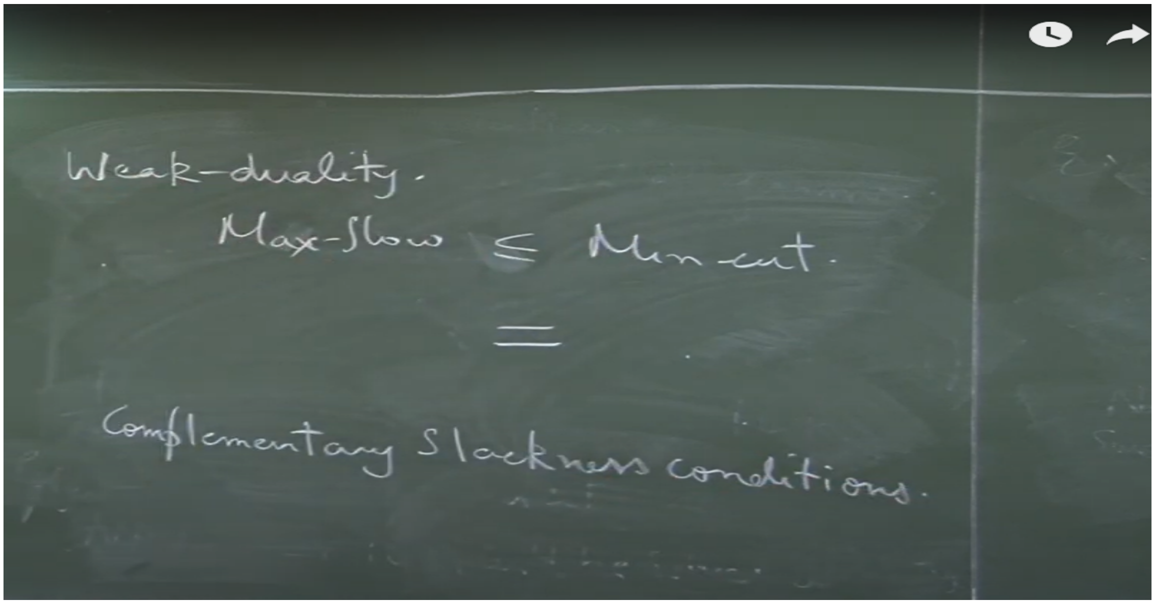
For Min-cut ILP = LP. So, let me prove this. Once I prove this then we will see the consequence of it. But once again a very simple property will be very helpful. What is generally easy for primal and dual? Weak duality, right. So, what does weak duality tell us? Even if we did not know anything about this, it tells us that max flow is less than min cut, right.

This is a proof of a very intuitive thing, right. You kind of can say that if you look at the graph, you look at the cut, the water has to flow has to pass through that cut at some point of time, right. I am not giving mathematical proof. The mathematical proof is here. But intuitively you can prove it mathematically also in different ways, but it is a very nice intuitive statement, right.

If you have a min cut, you cannot push more flow than the value of min cut. In some sense it has to pass through it at some point of time, right. So, then that means max flow is less than min cut. This just follows from weak duality. Simple nice proof of that.

What we will prove is that this is actually equal. The best flow is equal to the smallest cut. So, min cut is obviously a constraint, but in some sense that is the only thing which constraints you from putting more and more flow. Once again strong duality part is the interesting part. It is the more surprising part.

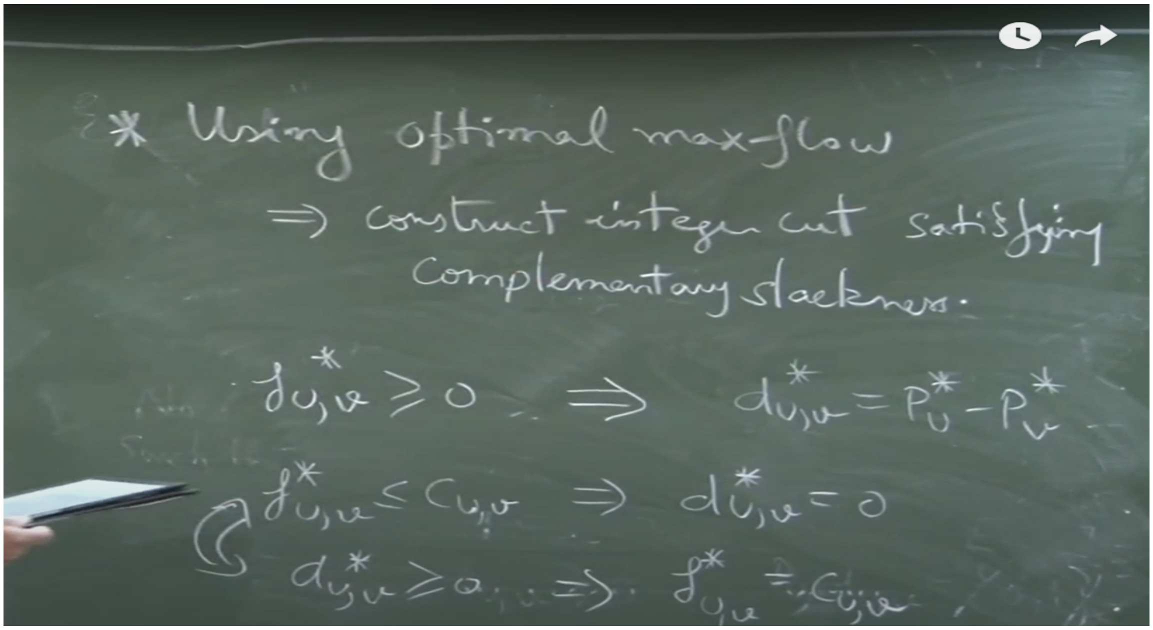
Weak duality comes out intuitively. Even the equality proof is kind of nice. We are going to look at complementary slackness conditions using the best solution for max flow and the complementary slackness conditions. We will consider a we will form an integer min cut solution,.



So, this is the idea. Using optimal max flow, construct integer cut. Integer cut means either one or zero. We are not assigning a wave a vertex. It is not saying that the value of the cut is integer. It is saying either the vertex given to the cut or not given to the cut.

I cannot give 0.5 of the vertex to the cut. So, construct integer cut satisfying complementary slackness. Then this integer cut and this max flow since they satisfy complementary slackness they are feasible. That means they are both optimal. That means even the LP has an optimal integer solution.

That means these two are equal, right. You should be very clear about the relationship of these two. What is happening is we are saying that this has much bigger feasible region as compared to this. So, when you want to relate them you want to say either that the optimal here also is a solution here or we want to say that if you take any optimal here you can modify it a bit make it a solution here and remain close to the objective value. In this case we are going to show that there is an optimal solution here which is actually a solution here.

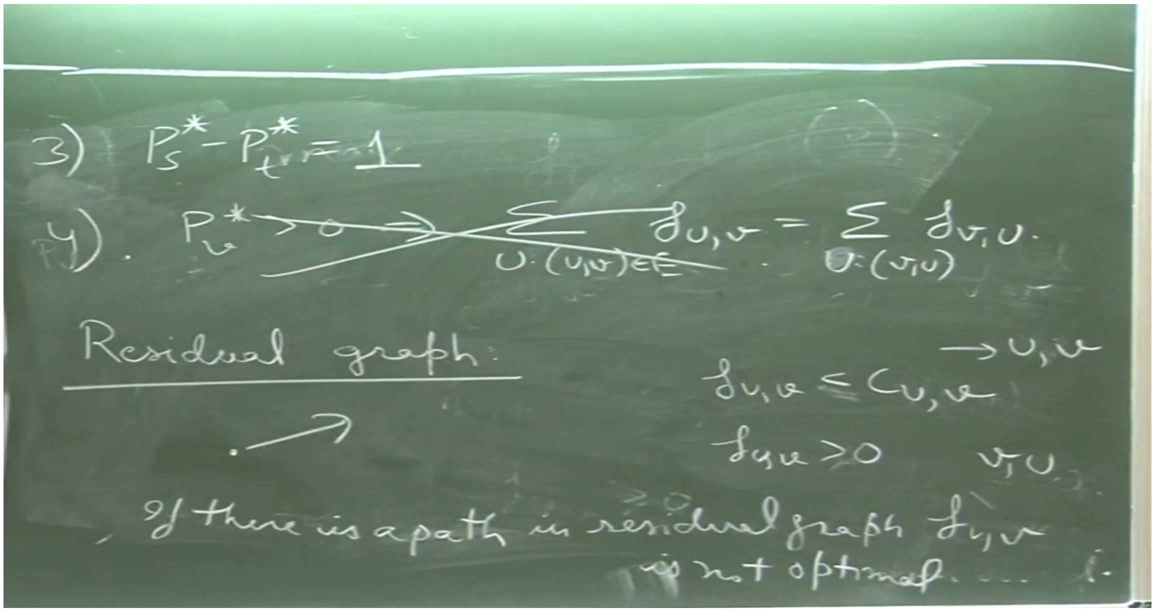


Let us write down the complementary slackness condition. Again by star I mean the optimal solution, ok. If this is 0 then the corresponding constraint should be tight. The corresponding constraint was correct. Or equivalently these are equal probably this was a better way to write.

So, this has to be 1 because we assume the flow from S to T has to be positive. Just give me a second, right. So, one thing which I did not mention probably I should have mentioned when you take the dual you might get stuck there. Remember this was our flow conservation constraint. If this is the constraint what is the equivalent variable? What is the variable for this? P V.

If this is equality, P V will turn out to be unconstrained. My claim is in my linear program for max flow I can replace this by less than equal to. Can you tell me why? So, you remember this is the constraint for all vertices in the max flow. I am saying I can remove equal and I can just say less than equal. It will still be the same linear program.

You have the intuition. So, just to make your intuition precise if at every vertex the flow if incoming flow is less than outgoing flow and the overall flow has to be conserved then everything has to be equal. And if you want to mathematically prove it you sum up this side you sum up this side you get the same quantities they are equal. If the sum of these quantities is equal then everywhere there should have been an equality, ok. Actually when you take the dual in the flow conservation constraint will have less than equal to. That is why I am writing this condition.



But this is useless for us because we know this will automatically hold true, correct. So, this will always be true. So, this does not give you a solution, clear. So, once again what we are going to do is construct a Pu star, P v star given f u v star so that it satisfies complementary slackness condition. This is done through what we call a residual graph.

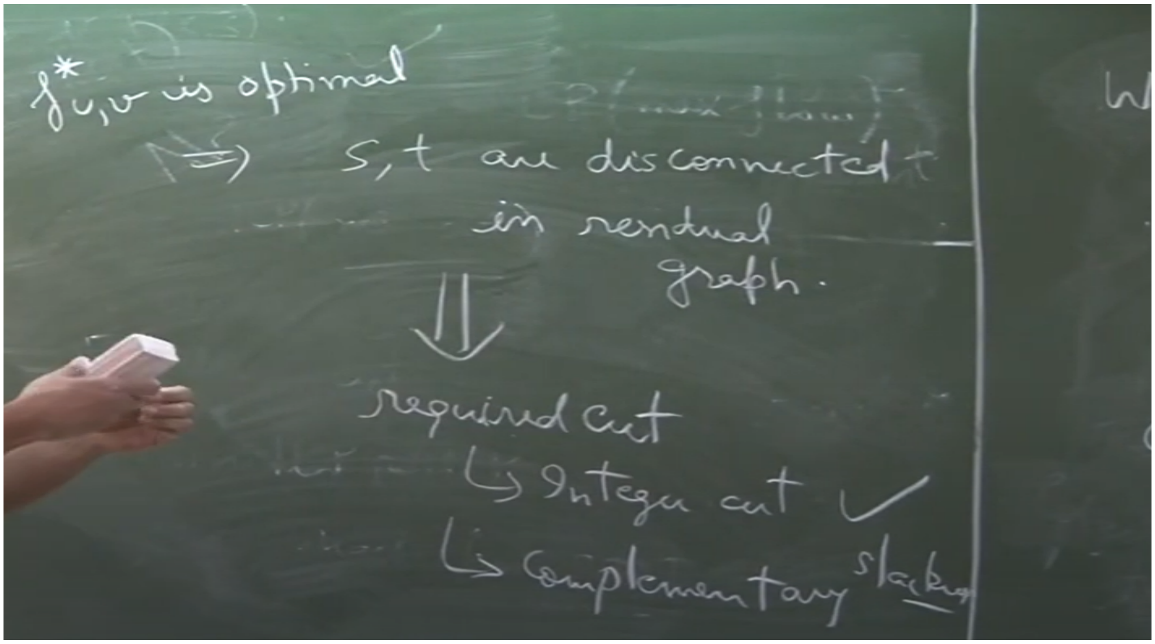
And this probably this idea was also introduced to you. What we say is that and this kind of emerges clearly if you want to look at a flow and you want to maximize the flow. You push the flow if you can still find a path where you can push more flow, right. Then the flow is not optimal. So, what do I do? I keep the edges. Oh sorry. If this is the case then that means I can push more flow through the edges.

So, I will keep them as edges u v. If I find the path of such edges clearly my flow is not optimal. Remember this residual graph is defined with the flow in mind. Once I have a flow I define the residual graph to improve that flow.

This is one. This is another thing. It could also be that I am going the opposite way with some positive thing. So, if f u v is strictly greater than 0 then I include v1u what do I know? Then can I improve the flow, right. If there is a path in residual graph then f u v is not optimal, ok. Other way to say this if f star uv is optimal implies, what is does implies? Other ways st are disconnected in residual graph, correct. What does this mean? This is specifying a cut.

This is what you wanted to do. Given an optimal flow you want to construct a cut. This gives us the cut. What do you want to check? It is an integer cut which is by definition. We are taking a cut and we are saying whatever is in the cut assign it 1. Whatever is the edge going from s to s bar assign it 1.

Everything else 0. But to show that it is optimal you want to make sure. That is much harder. What is the easier? What have been whatever I have been trying to do? Complementary slackness. I am going to tell you that this required cut satisfies complementary slackness condition.



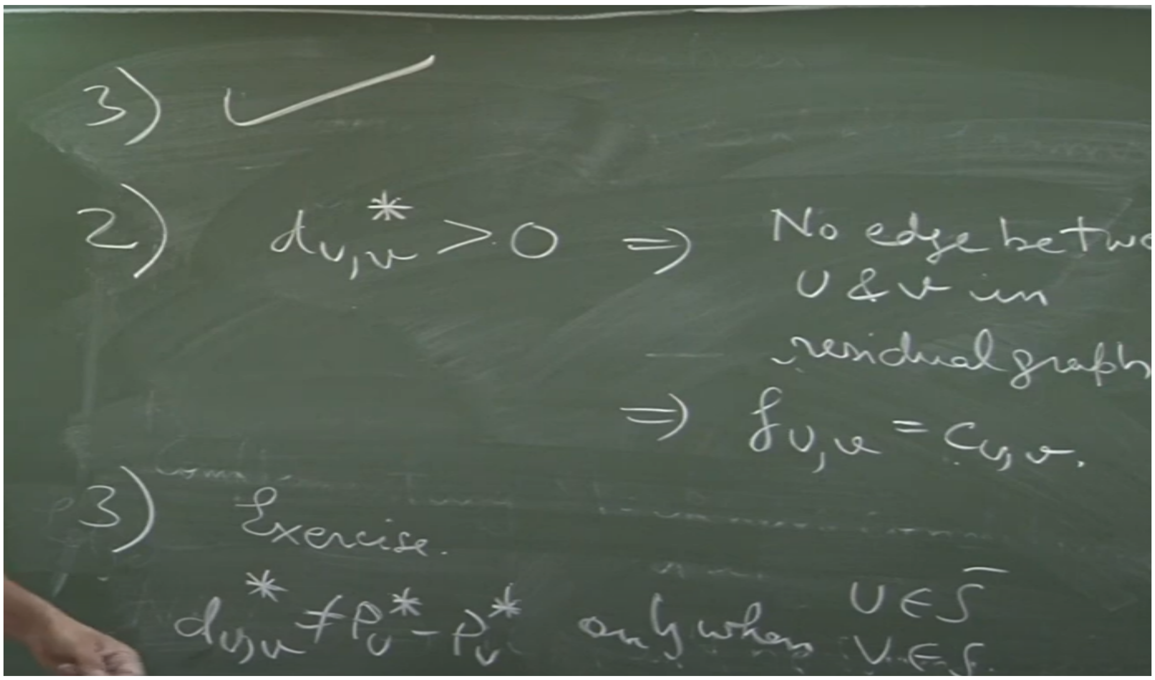
This is what you wanted to do. This was the idea. Using optimal flow construct integer cut satisfying complementary slackness. What is the cut? You take the optimal flow look at the residual graph. The residual graph has to be disconnected. It specifies a cut the set of vertices which are connected by s.

I just need to show that it is a it is satisfies complementary slackness condition. What is the easiest condition to satisfy? By definition since s and t are in the different 1 p s will be 1, p s star will be 1 and p t star will be 0. What about this? If d star u v is greater than equal to 0 that means in the residual graph there is no edge between u and v. That means f u v must have been equal to zero. This implies it also implies f u v is equal to 0, but we do not care about it.

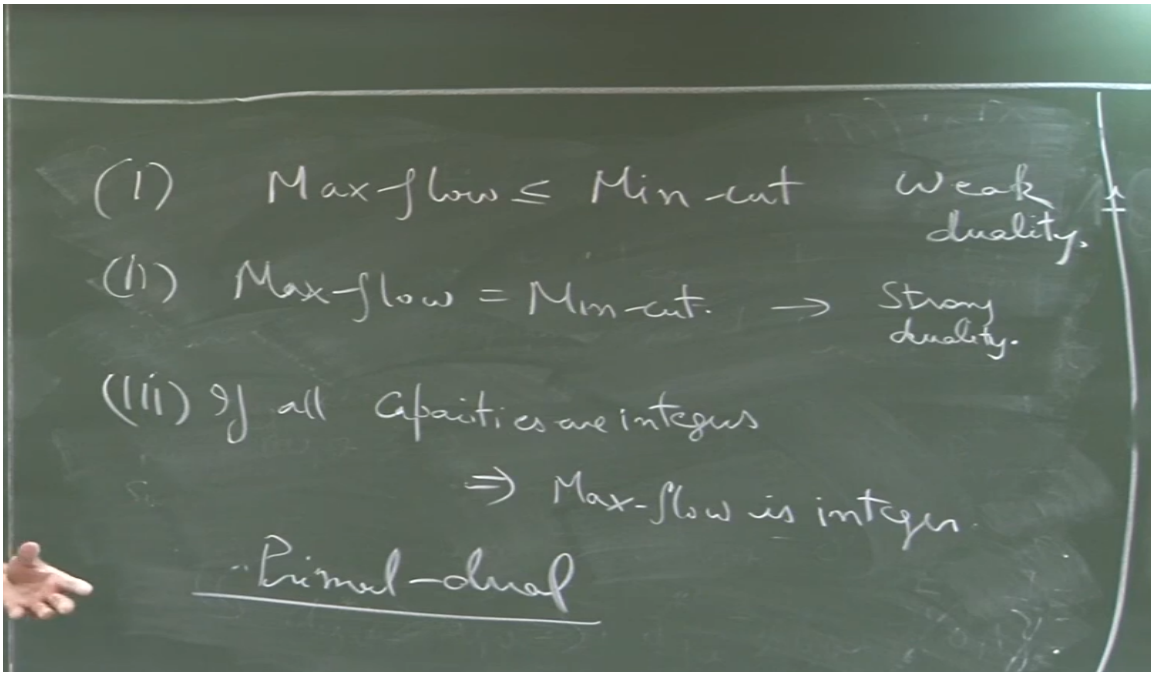
This also implies another thing it implies f u v if f u v was more than 0 then also I will get this edge, but I do not care about that condition. I only care about this, ok. Then now this condition this I gave it as an exercise. The thing to remember here is I can give you the idea.

Now if I want to show that this is verified, right. This means I want to show that if this is true this should definitely be true, right. This is Boolean logic a implies b is always correct except if A is true and B is false, correct? That is the only problem there. Now look at this quantity. This will all whenever u and v were very nice if s was in if u was in s and v was in s bar this will be 1, I will take it in the cut, fine.

If both of them were in s this is by definition of this thing. This will only be false irrespective of this condition this will only be false when this is in s bar and this is in s, ok. So, this is something which we have seen multiple times initially when we construct the linear program we said we can put greater than equal to for this case. But for this case again you can use the property of residual graph to show that in this case if this was the case then this would have been false because the property of residual graph.



Sorry, this should be greater than 0, and. This is complementary like this condition, good. So this proof you can do on your own, right. I have almost shown you everything last part also similar argument like this. So, again you might have seen this in algorithm course in different worlds. But what I am saying is by writing a problem in the form of linear programs it already gives us lot of information.



For example, directly we got, this came from weak duality, correct? Is equal to min cut, ok, right. Another very interesting thing we get is if all capacities are integers implies, righ, because if all capacities are integer then min cut is definitely going to be integer and this is not easy. This seems intuitive, but this is not easy from this we even get this. And notice here nowhere have we solved an LP yet. All these properties are just because of the structural properties of linear program.

This is something which I am highlighting from the beginning I am saying do not always think of LP as oh I can solve it efficiently that is one part of it. Most of the applications derive from the fact that LP has this nice structure. We know a lot about its feasible region. We know a lot about duality that gives us lot many interesting properties.

Solving it is I think a probably a very small part of it,ok. And now so this is one, but actually for this one we will actually see a solution. So, obviously you can solve it using lipside rather than everything. But in this case there is a nice solution which is called the Primal dual approach of solving a linear program. It works in some LPs we will see how this is done. And this will give us a known algorithm probably you have seen that algorithm in some other language we will write it in the linear programming language for the next class, ok. See you in the next class. Thank you. .