Indian Institute of Technology Kanpur

National Programme on Technology Enhanced Learning (NPTEL)

Course Title
Digital Switching

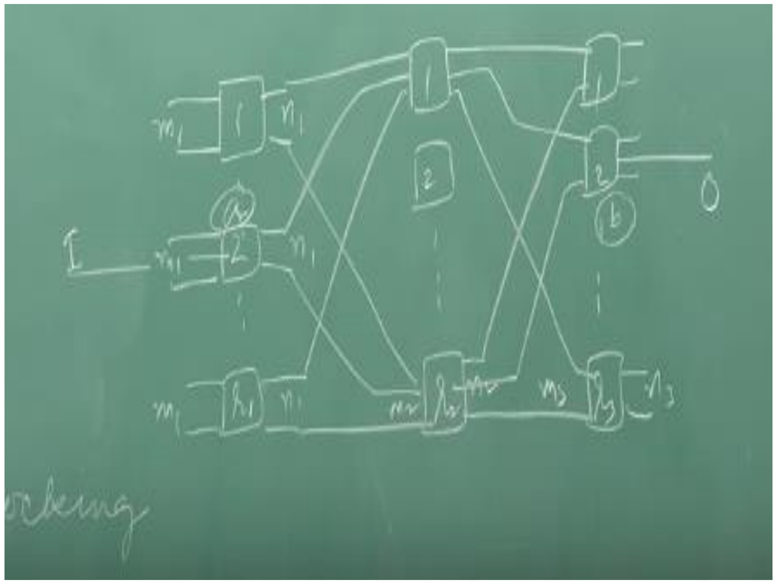
Lecture – 18

By
Prof. Y. N. Singh
Dept. of Electrical Engineering
IIT Kanpur

In the last video what we had discussed was a looping algorithm for a 4/4 or 8/8 ,in general for the Ben’s network .And we were able to actually generate mapping from any input to any output combinations . Any kind of map we take we were able to setup the connections ,but we have to carefully use the algorithm .If you recall I had also shown an example that there was a blocking in one of the switch and for the same map then, I could actually do what we call again another connection diagram using looping algorithm and call connections we were able to meet through.

So technically if there is the blocking happens you just actually dismantle all the connections, do it again as per the looping algorithm and you will be able to setup the connection that is the basically the re-arrangeability in the switch. So its a re-arrangeability in non blocking switch okay. So what we do is now we will actually generalize this motion of rearrangibly non blocking and we would like to understand in the Clos network under what conditions a Clos network becomes a rearrangibly non blocking so this is essentially is being.

(Refer Slide Time: 01:39)



Known as slipean theorem this is again for uni-casting we can actually do a similar exercise for multi-casting also okay. So at some point of time if time permits, I will probably will also introduce the equivalent if slipean theorem for a multicasting scenario, okay. So this theorem actually states that a three stage Clos network is Re-arrangeability non blocking ,If n only if so this is a necessary as well, as sufficient condition for rearrangibly non blocking switch so I will also draw Clos network to again ,emphasize on what this symbols actually means.

These are the same symbols which we have used earlier also so the r_2 which is the number of switches in the second stage is greater than equal to maximum of m1 is the number of inputs to the switches in the first stage so in a first switches are m1/n1 okay is under that is an n_3 stands for the number of outgoing inputs in the third stage switches so third stage switches are n3/n3 switches okay so this is the condition this is the Clos theorem and we have to essentially now prove that yes this is going to be true.

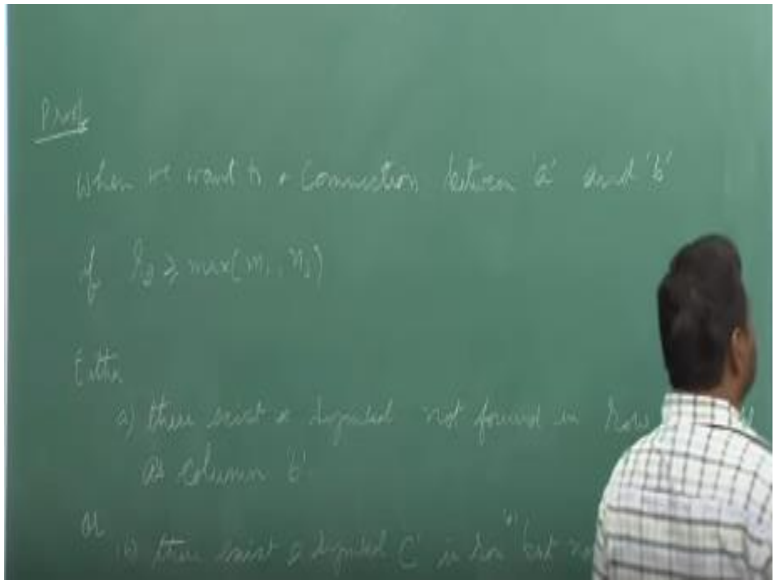
Now this is the particular case when the switches symmetric so I can further it in particular a symmetric network with m_2 = with n_1 = n_3= r_2 okay this is actually happens and another condition is m1 should be equal to n3 = we define as n so this is the condition for symmetric city this condition ids always through for the Clos network so in fact I need not I need not tried this I can actually only work with this condition so symmetric network with this thing is rearrangibly on blocking.

If n only if so this is the actually symbol use for refine only if r_2 is greater than or equal to n so in fact this is the this same condition de generates to this under that scenario just for recollection rearrangibly non blocking actually means when a switch is being operated upon you want setup a correction and there is a free input and there is the free output port but you are not able to setup the connection because there is no path available but if you can actually using in mechanism you can dismantle certain existing connections and again.

Create them same connection are being established again but by doing so you are able to create a path for this the free input and free output port which you are for which you are trying to setup the connection then switches are rearrangibly non blocking switch okay so let us see how we will actually proof it so we are looking into the Clos network so Clos network again for reference I am just drawing it here so this Clos network what we are talking about so in this case now let us start with the proof of this one.

So there is a free input port somewhere here this we call switch a this we call switch b okay so this is the switch a this switch b there is a free input port output port here and there is a free input port here so this is what we want to connect okay so when we want to establish a connection between and b.

(Refer Slide Time: 08: 52)



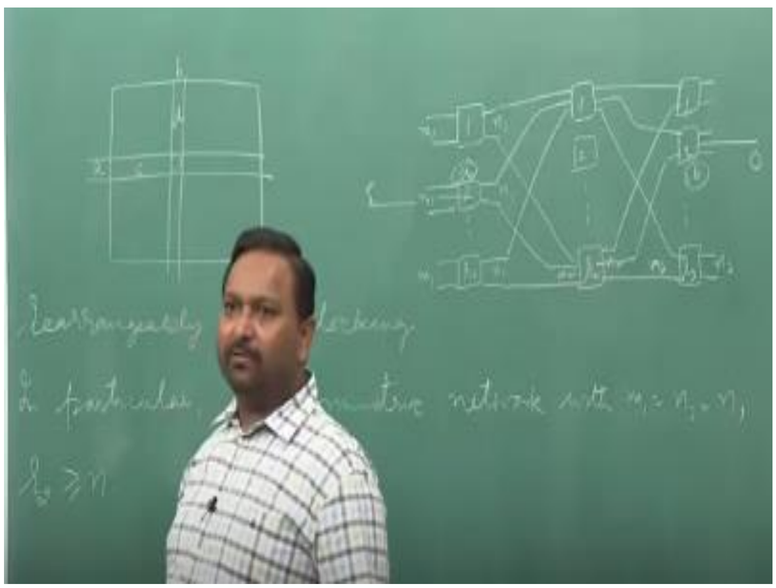
So these are the switches if r_2 is greater than or equal to maximum of m_1 and n_3 so if this condition is satisfied and we want to setup the connection between a and b a and b have been identified there in that case there will be two possibilities either so this actually setting of connection equivalently in a Paull’s matrix is equal to what if I draw a Paull’s matrix for this Clos network so a must be some row and b ahs to be some column so how we will put up a connection we will actually put up some.

Second state switch that entry has to be put in here okay now I should either find out a symbol so there are 1 2 the total r2 symbols representing the middle state switches so they can be here so depending on a has been connected to which all middle state switches these connected to middle switches will decide to what all middle say switches are represent here I need to find out among in this row as well as in this column some element which is not there in row as well as in column okay either that thing is going to happen.

So I will state that their exist a symbol not found in row A as well as column B or either this will be true but sometimes this may not be true in that case the second condition is always going to be true so this either and nor will be happening, we will actually prove yes this is going to be true, so the B condition will be that there exist a symbol C.

Which is actually in a row but not found in the column B in row A but not in column B also note it this will be happening, you should also n a symbol B so you will find a symbol C which is there in row A but it is not in the column B, okay. So there is some C here but it is not present in the column and you will also find but it is not present in the column and you will also find a symbol D which is in column B. But not in not found in row A, okay.

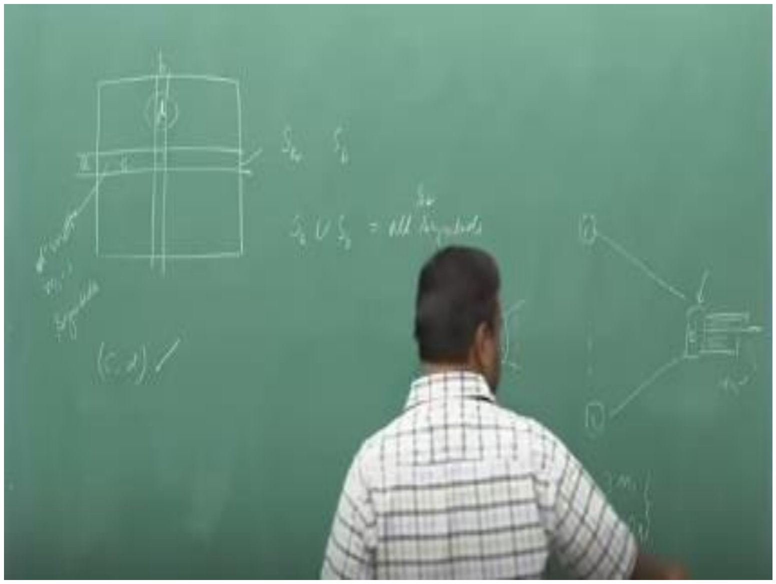
(Refer Slide Time: 13:28)



So there is also a going to be a symbol D so C and D the pair will be existing, so one of the two thing should happen either this should be true or a pair should be happening, okay. So let us see whether this argument is true or not if this is true then I will be able to identify a rearrangement mechanism, okay. So let us see how this will be happening.

Now of course we do not need this we will be coming to the example later on that I will draw it again, we require the pulse matrix.

(Refer Slide Time: 14:26)



Now if condition A is not true, what is the meaning of this, if condition is not true it means if I look at all the symbols in A all the symbols on B and I create a union of this symbols you know of these set all symbols have been consume all R2 symbols the middle stage switches of the Clos network have been consume they have been put in row and column both.

That is why you are not able to find out one element which is not there in row as well as in column B, okay. So that what must have happen but if that is true that number in fact symbols here in row A. I can write as A set of symbols in row B I can write it has row as B, as A U B consumes all symbols, okay. Is equal to all symbols but remember when I was I would actually drawn the A there I had drawn the B there, there was one output which was free there was one input which was free, which you where trying to connect, okay. So how many symbols would have been occupied?

So this output is being connected to 1, 2 R2 symbols it means and remember this R2, R2 is greater than M1 and R2 is also greater than and 1 and 3, so when R2 > = M1 the number of inputs which are occupied are in worst case can be M1 – 1 you just forget this one so remaining once except this will be M1 – 1 which actually implies when in row A there can be only M1 – 1 symbols in worst case.

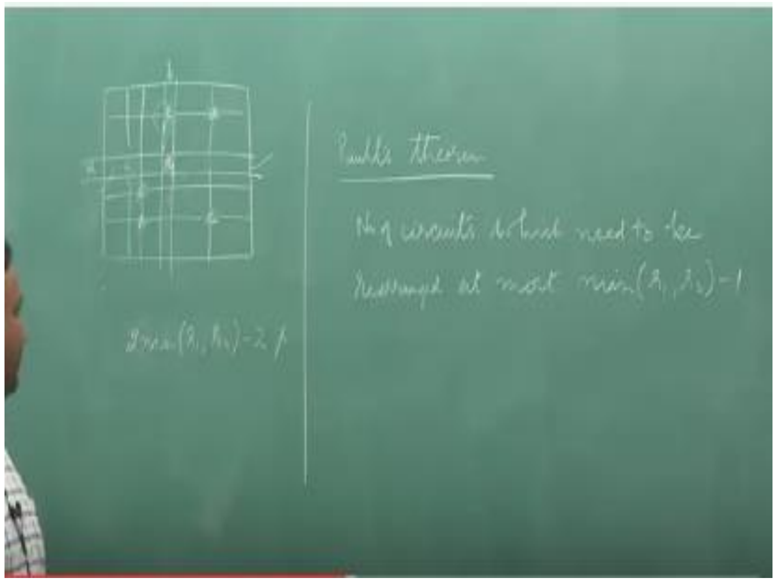
This is because you are having one input which is free which you where trying to connect, okay. And your R2 > M1 it means there must be a symbol out of these R2 symbols which is not present in row A but my condition say that is take these two union of these two sets I am consuming all R2 symbols, so there is something which is missing here but which is present here, okay.

So which is going to be D, okay there is something which is missing here which must be present in column, so this element we call it as D so D is not present in row but it is present in column, I can take a similar logic from this side so maximum number of all middle stage elements which can be connected from B can be only equal to number of output channels output ports which are occupied which is M3 - 1.

One is this is free which you want to connect, so which actually implies the maximum number of symbols in column B has to M3 -1 but R2 is > maximum of these two is greater than both of these two actually which implies there has to be something, so all R2 symbols are not in the column but since union is consuming all the R2 symbols there has to be something which is not present here but which is present here.

So that is C, so I will always be able to find out a pair of symbols call C and D, okay. So this C and D symbols CD pair either I will be able to find out this because case B is true or I will be able to find out a symbol which is not there in row as well as column, so one of the two conditions are going to be true. Now worst these are true more off these two will be true, how the rearrangement is going to happen. So let us see, so consider this Paul’s matrix.

(Refer Slide Time: 19:13)



So in case the condition 1 is true you can find a symbol which is not in row as well as in column I can just take that symbol and put it here, so that symbol you say E I will just put it and I can use it to set up the connection from the free input port to free output port but if condition 1 is not true the condition 2 is going to be true, one of the two is always going to happen.

So we have just actually discuss that thing, so if the condition two is true I can always find out the C here and a D here, so now what to do, okay. The way thee rearrangements will be happening is I will look into the row where this D is there, okay. So remember there is no element which is duplicated I am talking about a uni- casting connection, okay. All everything is uni-cast there is no multi-cast.

So there can be element D cannot repeat anywhere, D can there can be only one D in this space, okay. So this actually means I have to now look for here some wherever there is a C which is available remember C was not there CC actually can be there anyway CC was not there in the coulomb C was present in this row but not in the column and D was not in this row so it was in the column.

So I will look for a C, if I can find a C it is fine, if I cannot find a C so there is not C in this row then life is simple I can just put a C here and put a D here connection is set up, so this a rearrangement, okay. We will actually take an example for a general for a example network Clos network and we will try to see how we will do this, we will use Paull’s matrix there to figure out the rearrangements. If you end up in finding a C then of course you cannot put a C here because C will get duplicated, so then what you have to do, you have to look for a column.

Now will you ever reach and find a element here a D here no you could not find it d here because d was not there in the row so you will exclude this in the search you will not looking to this row everywhere you will be looking and you may enter pin finding a d here. You can keep on doing it you can next time look for a row try to look for a C.

Now can you find a C here you cannot because C was not present here in the column okay you will not be able to get a C here in this particular column because C is already present so you will be always looking for the columns here this, this and this column will be exclude three of them so somewhere else you may get a C, so you will key this is what we call chain search you are doing now chaining.

You can keep on doing it so next time you want to find out a D you cannot get a D here you cannot get a D here you cannot get a d here everywhere else now you are search is space is reducing every time. Okay you may enter pin a D somewhere and then you will try for a C so you cannot get C here you cannot get it here you cannot get it here or you may not get it chain probably can end here.

In worst case what will happen is you are doing search and ultimately when all columns are all rows are apostate you can know more move any further and the chain will end so chain is going to end is guaranteed okay because every time when chain is being extended it going to a new column or new row so once you cannot extend chain any further you have to simply replace all d/c and all c/d and put the d here and you are able to set up the connection.

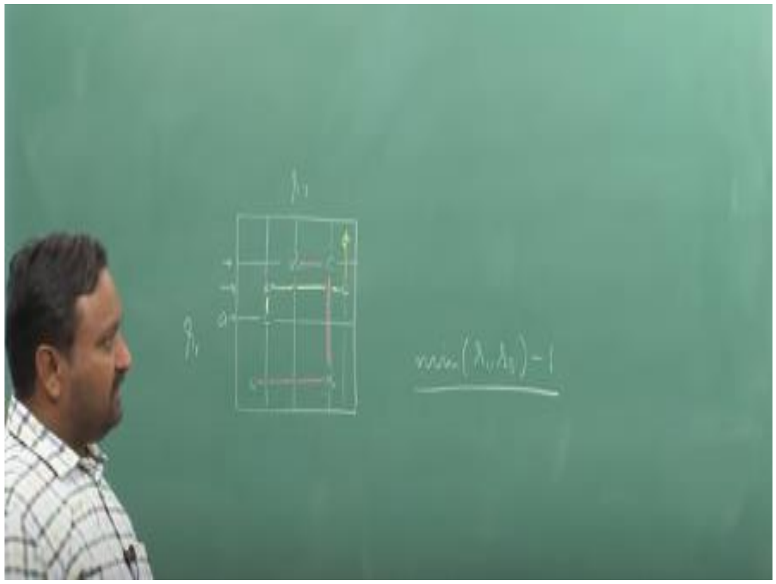
Okay so when we actually doing this search this row and this column are being excluded actually and every time you are doing one row and one column so in worst case so in the beginning people thought this a very simple thing that R1 + R2 which are the number of rows and columns -2 in worst case these many re arrangements will be required so these are the re arrangements which are there and then finally you will make an entry in this place.

So this at most maximum but this is not the correct bound in fact every time you do row and search the rows actually finish first then you will no more get no more be extending and you can actually prove it through a very smart way so if the row 1 R1 is less or R2 is less is the all rows are finished even if the extra rows are there you will not be traveling there so in fact this should have been put to two minimum of R1, R2- 2.

Okay that put the at most number of re arrangements which are required this is further improved by something known as Paull's theorem so the Paull's theorem says so we have actually proved the slip ion to this theorem so far okay only we are we are actually able to prove that if R2 is greater than maximum of M1 and M3 one of these conditions will be happening and I will always able to set up the path, okay.

And that is a good enough condition to prove it so Paull's theorem is states that number of circuits which it to be here is not this so number of circuits which need to be rearranged at most will be minimum of R1 R2 – 1 is not twice so this number was twice actually original in the theorem so Paull's theorem actually improves on this. But what is the premise how we actually get it? So the idea here is very simple I can actually describe it here.

(Refer Slide Time: 27:17)



So what we do is we take the same Paul's matrix. So there is a and there is a corresponding element c in this case so how to determine the rearrangements I said first of all find out go across there will be a somewhere a c fine but I was only extending one chain now let us do it alternatively so next time I will look for a column here find out if there is a d somewhere here remember this is excluded when I am searching for D okay.

Now once I have got this chain next I will look in to this and find out a d somewhere this d cannot like this is including this particular row because d cannot be duplicated in the row. Okay so I have extended I have let me use correct chocks here for clarity so next I have extended this chain by two steps now I will use this I will do alternate chain extension I can go like this and get c somewhere remember I will not be revisiting this particular column c is already there present so it cannot be duplicated c was not here and this will be excluded.

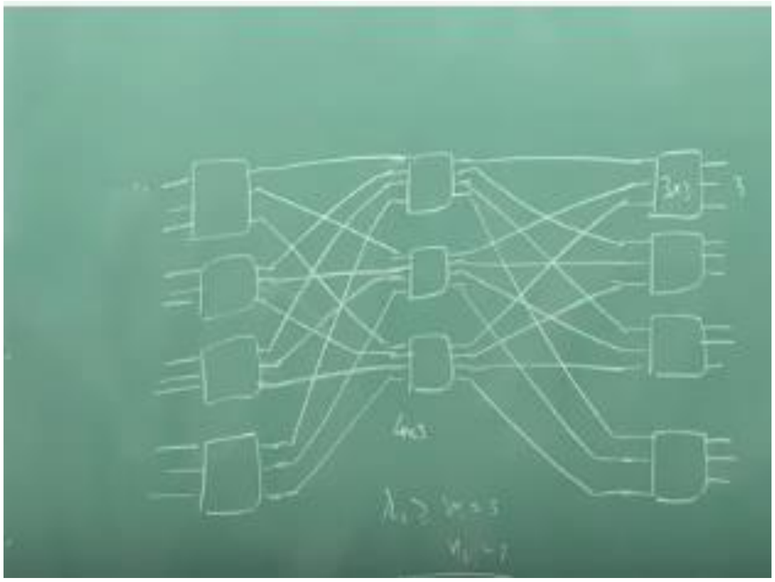
And then I will again go for the next chain so I am looking for c so will be excluded in this particular column this one I will be now looking for somewhere a here may be c will come here and then of course we will be looking for d you cannot come to this particular row this row is excluded this is excluded somewhere here maybe so you keep on doing alternate search and every time you are extending the chain by one step.

So when all rows so whichever a minimum of R1 and R3 whichever is minimum of these two that will exhausted first and one row and one column we are excluding in the beginning itself there depending on they are actually corr3responds to a and b so in worst case you will be requiring minimum of R1 R3 – 1 re arrangements for setting up of the connection in the clos network.

So this is what the Paull’s theorem is all about now what happens when you are using a rearrange able network actually increasing your circuit set up complexity in case of a stately non blocking switch one step you just simply set up the path here in worst case whatever is the value minimum value of R1 and R3 those many circuits will need to be rearranged worst case that is possible so that is your circuit set up complexity.

Which will be roughly R1 or R3 but if switch is not heavily loaded this number of rearrangements might be actually smaller in that case okay so let us do an example of this thing let me create the three stage network and a Clos network with this same conditions satisfied in try to set up the connection to justify that this is this how actually done to give an example. So let us do it.

(Refer Slide Time: 31:34)

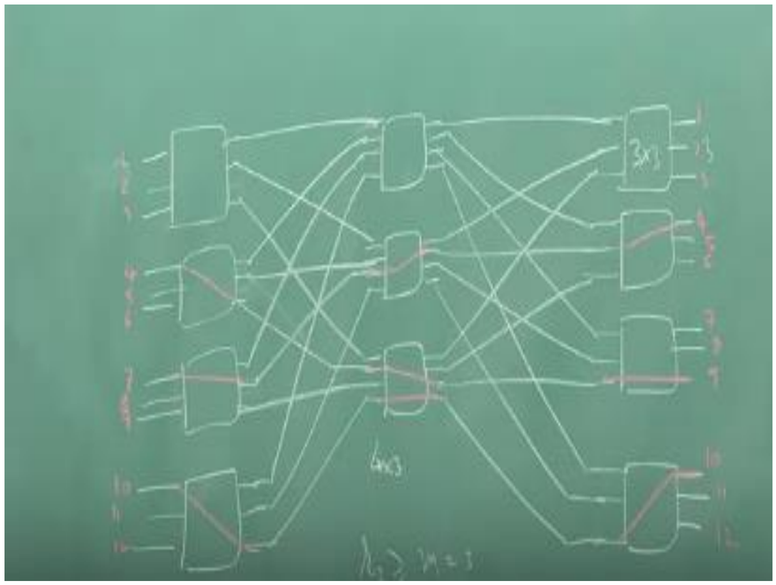


So let us take up a network I will take up a 3/3 and I am actually not take I am taking a symmetric system here intension sly give you an example or we may actually do it for a symmetric system first and then for a symmetric I can try that so let us take 3/3 so my output side will also by 3/3 so why the second is three because this is N3 is 3 M1 is 3 M1/ N1 so R2 has to be greater than or equal to M1 or N3 whichever is the maximum of the two.

So in this case that is 3 so the middle stage is stage will be 3 also so this is a 9/9 switch so in fact I can actually generalize I can even make it 4A does not matter I can make 12/12 okay let us do it so number of switches in the middle stage will always be 3 okay so this will be 4/4 this will be like in connect, so this is what the switch is all about, these are regenerative switch, so these are four by four, three of them, r2 here is greater than or equal to m1 or m3, both are three actually, so r2 has been taken as thrice, so this is the regenerative switch. So let set up some arbitrary connection and see how will take care of the block.

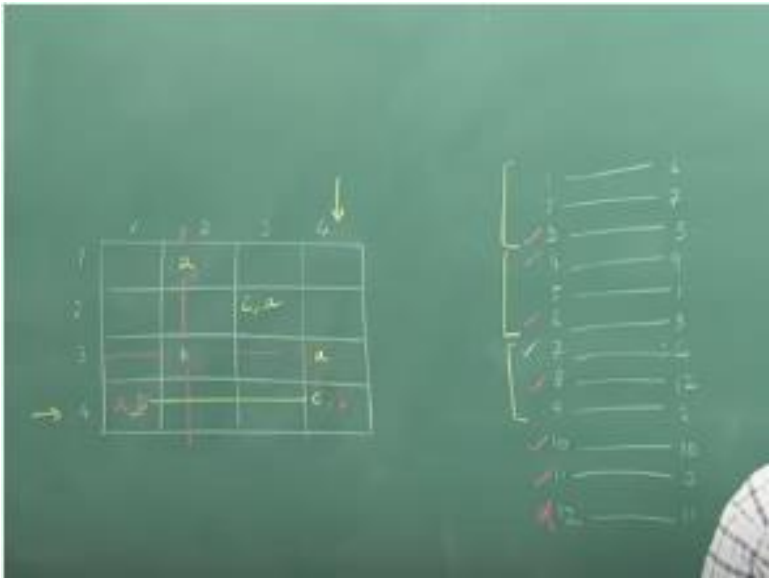
So I will keep on setting up the connection whenever the blocking will happen, I will try to use Paul's matrix to solve it, and then we will actually look at looping algorithm case and prove that this is nothing but exactly what I'm doing here, the same algorithm will work even in that case, so let's take up an example, map here, so this 1 to 12 at the input, 1..2 and I will not do it in order, I will just do random thing.

(Refer Slide Time: 35:30)



So this we connect it to, just take some random stuff, so that's what will be the map, so I have just taken randomly something, let's try to sort out and I will do it in random order, I will set up the connections, so let's do with 7…4 first, I will draw it with different colour chalk so that you can actually see it, this 1, 2, 3, 4, 5, 6, 7. So 7 have to be connected to 4 and I can take arbitrary anything. I don't bother, whenever the blocking will happen, that time will correct it.

(Refer Slide Time: 37:48)



So let's put something, here it goes, 7 to 4 connections has been made, let's take 10 to 10, so far I have not seen blocking, 4 to 9, let me put 6 to 8, 8 to 12 let me put that, through this root I cannot do it, through this root, 3 to 5, if I don't get blocking, I will do the blocking later on, swipe the maps actually, 11 to 3 it's too power this will be done. 12 to 11, now I cannot connect from 12 to 11, this is not, this blocked.

Because if I want to setup 12 to 11, I cannot use these two, these are already occupied, to use this one, this already occupied, so how to connect 12 to 11 these are blocking, so now let's draw Paul's matrix and then rectify the problem, to do the re arrangements .In fact there will be pear of connections you can identify, which are blocked. So you have 4 by 4 Paul's matrix, so I will give the number 1,2, 3 and 4…1, 2, 3 and 4….1, 2, 3 and 4 and the middle switches I can always write as a, b and c.

So now let's look at the connections which we have already setup, and in the end I will look into this how it will be done, so I have connected 3, look at the top 1 by 1, 3 to 5 via a, so 3 to 5 via a, so a has to come here, next 4 has been connected to 9 via c, 4 is connected to 9 via c, then 6 is connected to 8 via a, then 7 is connected to 4 via b, 8 is connected to 12 via a, then we have 10 is connected to 10 via c,11 is connected to 3 via b and then i was trying to connect 12 to 11, so 12 actually comes in this and I am trying to connect here okay which actually means I have to make some entry here now look at the condition and look at all the columns A and C, B and C, if I put take the union ABC all three of the elements have been consumed.

Condition B was is being satisfying here okay but now how to do the rearrangements so let's see how we will do this before we can sort out this particular problem, so I need to find out if condition 2 is true, condition 1 is not true so condition 2 is should be true, so if I look and this row i found a B which is not present in column and corresponding there is A which is not present in the row.

The pair is now A and B so how to sort out this issue so one way is i can look into this B and then make the change I can try to find out the change that way or I can do it the other way round ,so may be let's do it this way so this is A let we find out the chain here okay so once I find out B here I have to look into the column so now if I do any change chain actually stops so what I have to do is I have to now represent simply replace this by B this B by A ,this by B and now A is free I can use A here .

So this is the re arrangement so if we do the rearrangements if the connections happens or not there is an another alternative which is possible we will also do that that requires that less number of rearrangements okay so what I have to do is ah from element 1 which is 3 to 5, this particular connection 3to 5 which was via A has to go via B so 3 to 5 this particular connection which was going via A I have to move it via B now.

Sp let we take it with so this connection also will be dismantled and i have set up this connection the one which i have dismantled was from 3rd to 2nd via B this one so third to second actually 7 to 4 which is this and this is what I have dismantled here so 7 to 4 now has to go visa A okay.

So which actually means this one will be dismantled so I will be connecting 7,4 via A and then we had a connection here which corresponds to A to 12 which has to go via B so A to 12 dismantled here it has to go via B from here A to 12 so it has to go downward now, so this being dismantled from upper actually 1 so this from here I have done 8 to 12 and now I can use A to set up the connection from 12 to 11 okay so now I can use 12 ,11 connection done I have done the rearrangements .by could doing a search here only B would have been move to A and then I could have been putting in A here okay.

So let us see if that also would have worked so I am actually now putting coming back to the original configuration so this will be A B A so now what I have to do is I have to replace this B by instead of doing through column I would have done it this way only one there is change search is stops I could have converted this to A and a B could have been put here .

So that would have been so I have to now reset the connections here ,so I will just set up these things again so 3 goes to 5 via A then 7 goes to 4 via B 8 goes to 12 via A ,11 to 3 so this is the current connection 11 to 3 is via this ,so i have to do that 12 to 11 so alternative is this particular connection this one has being dismantled this corresponds to 11 to 3 which has to go via A so 11 to 3 I will just move it via A .

Once this goes this one becomes free 11 to 3 via A okay so once that has been moved from via A then I can set up to 12 to 11 via B okay so 12 to 11 now can be setup via B only one re arrangement was required and a fresh connection was set up okay .So this was the more optimal scenario so minimum of the 2 has to be taken.

Acknowledgement

Ministry of Human Resources & Development

Prof. Satyaki Roy

Co-ordinator, NPTEL, IIT Kanpur

NPTEL Team

Sanjay Pal

Ashish Singh

Badal Pradhan

Tapobrata Das

Ram Chandra

Dilip Tripathi

Manoj Shrivastava

Padam Shukla

Sanjay Mishra

Shubham Rawat

Shikha Gupta

K. K. Mishra

Aradhana Singh

Sweta

Ashutosh Gairola

Dilip Katiyar

Sharwan

Hari Ram

Bhadra Rao

Puneet Kumar Bajpai

Lalty Dutta

Ajay Kanaujia

Shivendra Kumar Tiwari

an IIT Kanpur Production