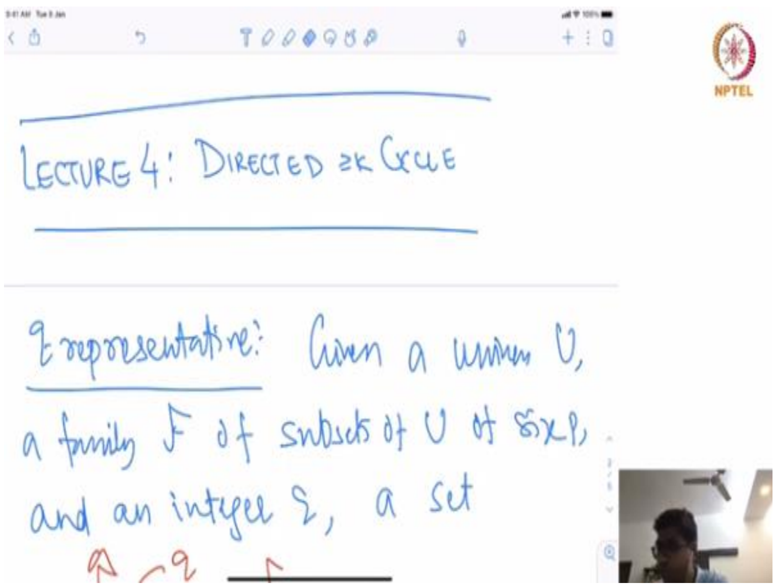
Parameterized Algorithms
Prof. Neeldhara Misra
Prof. Saket Saurabh
The Institute of Mathematical Science
Indian Institute of Technology, Gandhinagar

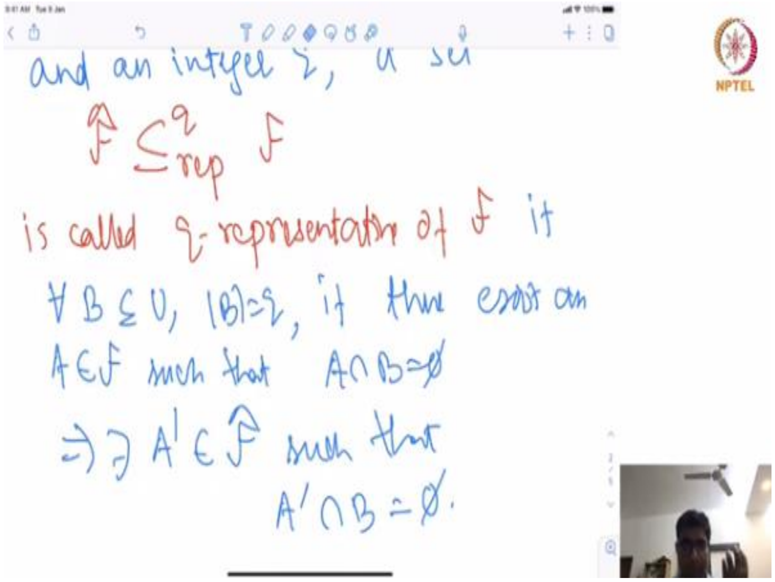
Lecture - 46
Matroids: Representative Sets- Applications (Directed Long Cycle)

(Refer Slide Time: 00:23)



Welcome to the fourth and final lecture on representative sets or matroids. We will give you one more example in this lecture about directed case k at least k length cycle taken and then tell you what are the generalization. So, first let us just do a recap of what we have done.

(Refer Slide Time: 00:35)

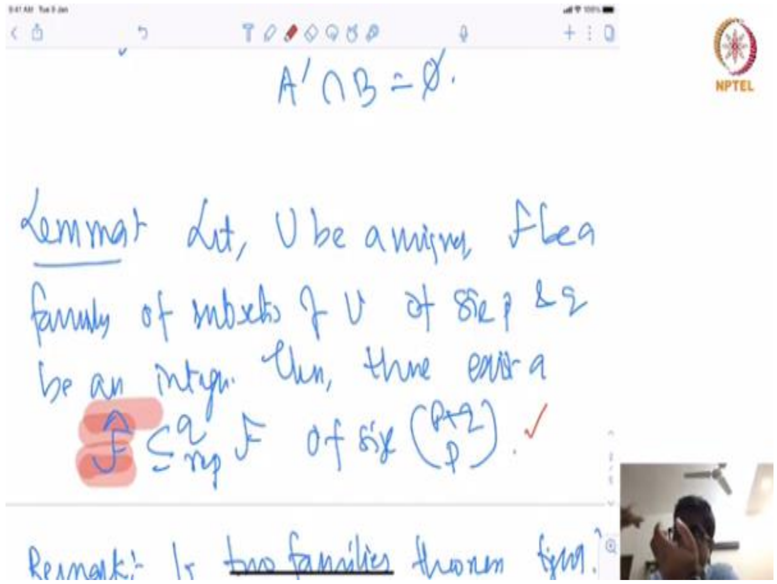


So, you are given a universe U, have a family F of subsets of U of size P and integer Q. And you want to produce a set a pad of f which is called q representative if the following property holds. What is the property holds that for all B subset of you, whose size is Q. If you are able to find a set A in your family, such that A intersection B is empty set then in my sub family effect also I should be able to find you a set A prime such that A prime intersection B is phi.

So, these kinds of sub families are called q representative, because it is able to maintain disjointedness property of f in f prime with respect to all sets of size q. So, this is why it is called q representative it represents f with respect to every q size set in terms of disjointedness. If you give me a set if that was, if you give me a set of size q if it was disjoint with respect to some set in the family F, it is disjoint with respect to some other set maybe the same set.

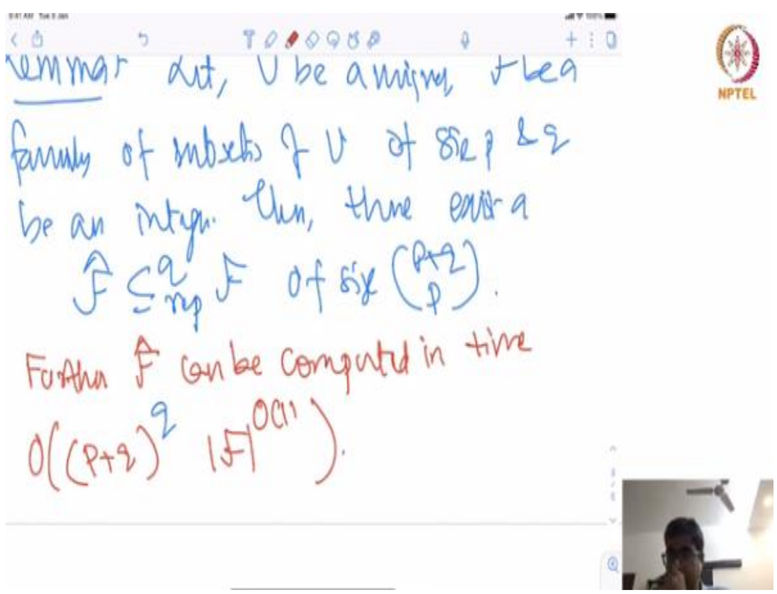
But in the family F hat. So, in the smaller family also I am able to produce a set which is disjoint from the set given. So, that is what the notion of q representative was.

(Refer Slide Time: 01:53)



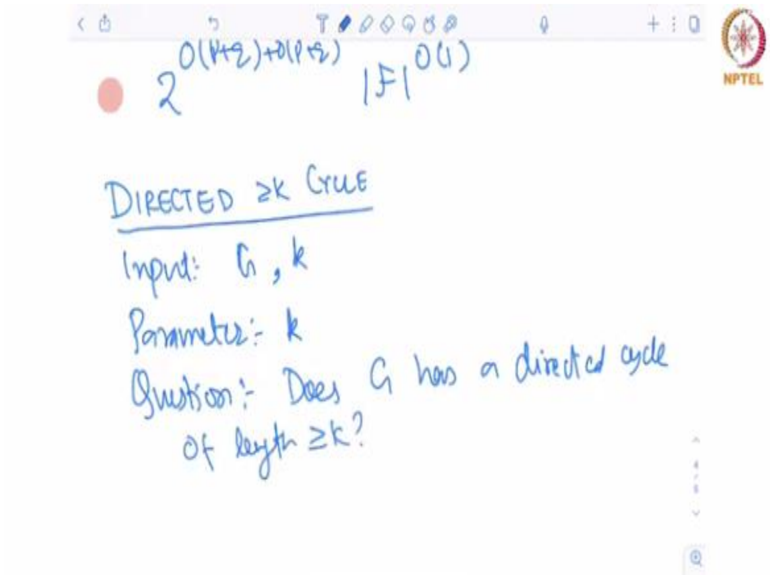
And what is the lemma that we proved. We proved that if we look at a minimal family of sets and we define minimal then there exist if hat whose size is independent of the size of the universe, it only depends on the size of sets in my family and the integer Q or the sets with the size of the sets with respect to which we are trying to call representativeness. And that was P plus q to phi.

(Refer Slide Time: 02:27)



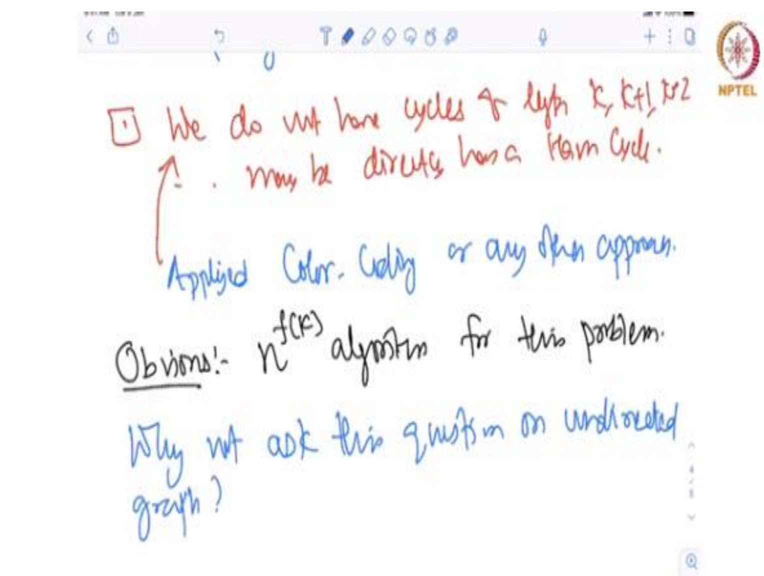
And further, we also talked about computation that there is an algorithm which can compute these things in P plus q to the power q if we go for. There are as I said, there are much faster algorithm for these problems and they can run in time two to the power big O of P plus q little o of P plus q. So, there are algorithms with these running times and they are the ones which are used to give faster and faster algorithms for some of these objects.

(Refer Slide Time: 03:07)



So, one interesting cute application that I would like to talk about in today's lecture is about directed K cycle. So, let us try to understand what is directed K cycle. So, input is going to consist of a directed graph G positive integer K, parameter is going to be K and the question is does G has a directed cycle of length at least K.

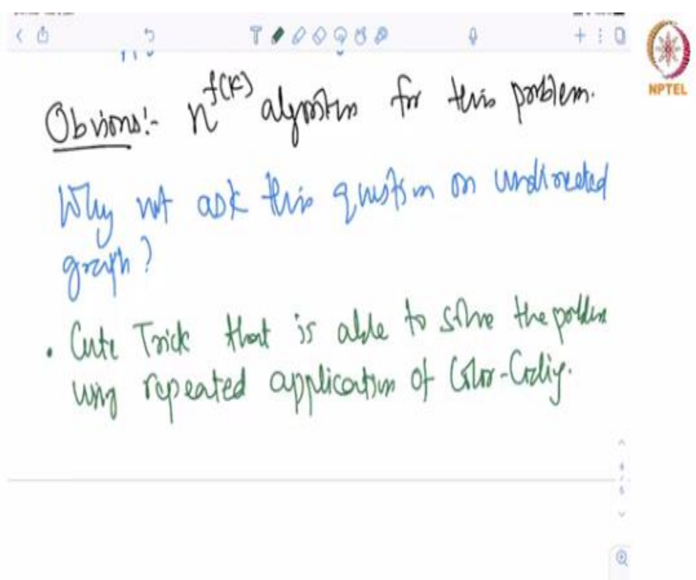
(Refer Slide Time: 04:12)



Notice that so, why the question is much more interesting that it could be that we do not have cycles of length K, K + 1, K + 2 dot dot dot but maybe directly has a ham cycle. But all we are saying to you that look. So, if you had a cycle of length K or K plus one or this in this case we could have applied colour coding or any other approach. So, in that sense this question is very different that G data directory cycle of length at least K.

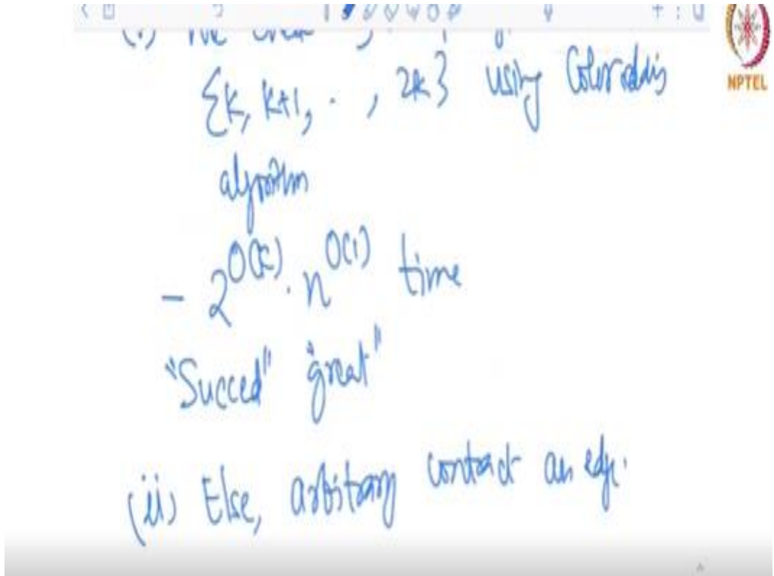
So, we want to test for that. So, it is possible that graph does not have a cycle of length any function of K, K plus but it has a much larger cycle. How can I find such a cycle? It is not even obvious. It is not even obvious that we can design n to the power f of K algorithm for this problem. You may ask why directed graph if the way the condition. Why not ask this question on undirected graph. Valid question, yes, this question makes sense even on undirected graph.

(Refer Slide Time: 06:12)



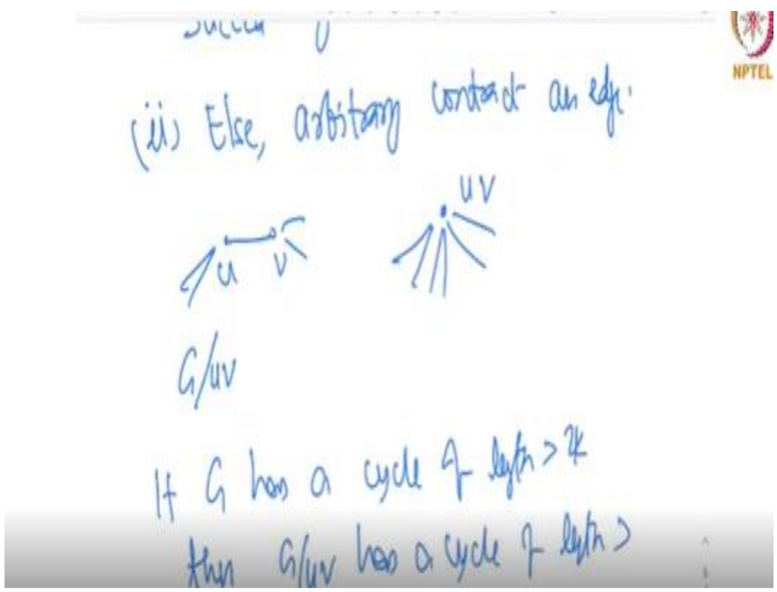
But for undirected graph there is a cute trick. There is a cute trick that is able to solve this problem using repeated application of colour coding. I will tell you that you wait you pause the video and then think for a few minutes before looking at what I did what I do next.

(Refer Slide Time: 07:00)



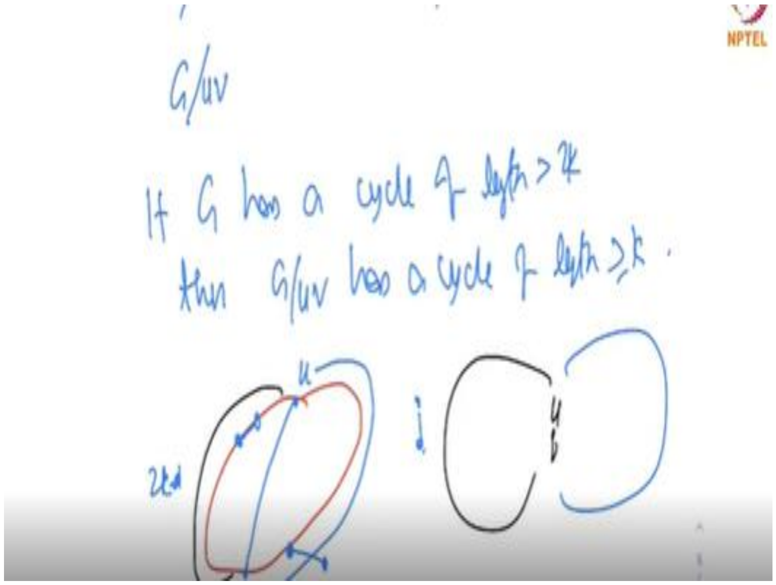
So, the algorithm is very simple for this for undirected graph. The algorithm is first raised simple is that look we check cycles of length K, K plus one up to 2K. Just check whether do we have a cycle of length between this using colour coding algorithm today. And this is going to take you some two to the power of K and to the public have done. So, if you succeed great. Else what do you do? Arbitrary contract an edge.

(Refer Slide Time: 08:11)



So, you pick up an edge and contract so, what is a contraction means the contraction means basically if you have to vertex and edge uv you make a single vertex uv and make everybody which was adjacent to you, you make adjacent to uv. So, this would contraction me. Now let us look at G contracted you will see a symbol that we have. Now I claim to you that if G contracted uv, if G has a cycle of length at least greater than 2k then G contracted uv has a cycle of length greater than or equal to k. Why?

(Refer Slide Time: 09:18)

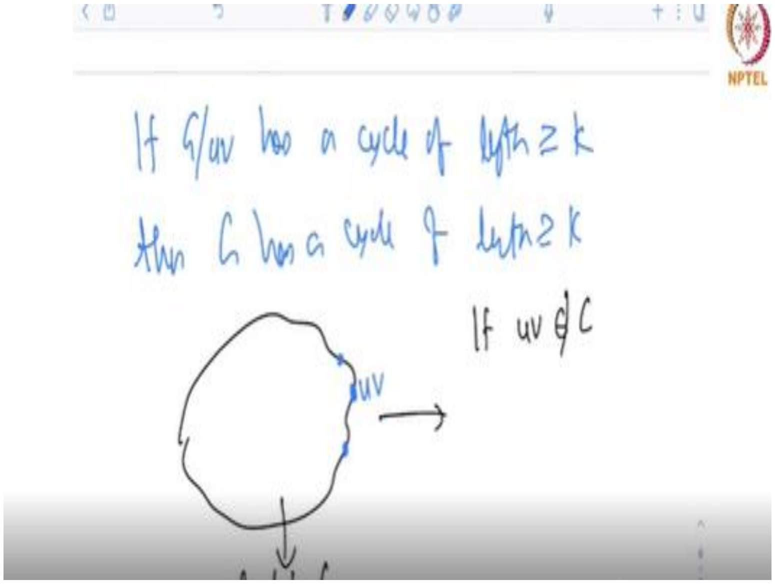


So let us prove this. So, if you had a cycle of length at least 2K. So, here the cycle. It is possible that we have not contracted any if we just contract and edge on the cycle then that is great. I mean, there is nothing to worry this cycle will survive. If we contract an edge completely outside on the cycle, nothing will happen this cycle will survive. What I mean by this if you contract this is the length of the cycle is then like 2K took at least 2K.

So, now you have greater than 2K so who is like still more than K. And similarly if outside nothing will happen. If you contract and nothing will happen. The problem could be that I could have a chord I could have a chord like this say this is u and v. Now, if you contract uv what will happen. If you contract uv notice either this part of the cycle or black part of the cycle. So, now you will get to cycle it.

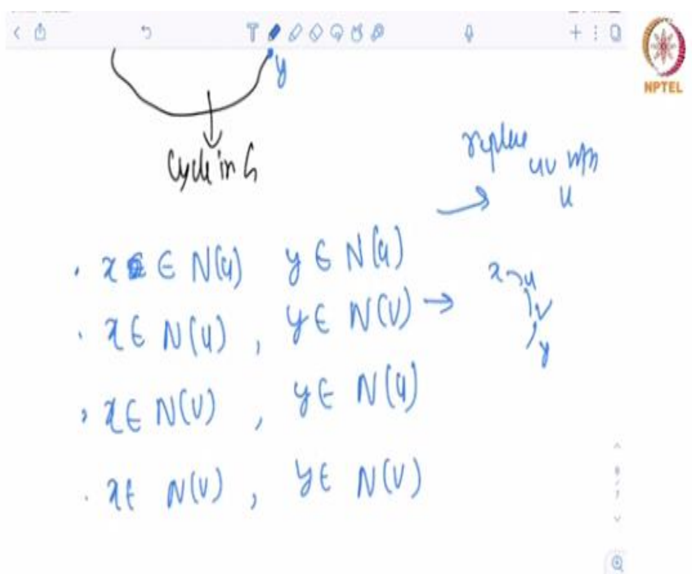
If you contract up you will get a cycle starting at u ending at v are the other cycle. The blue cycle is starting at you and ending at v. Because how the picture will look like it will become uv identified and black cycle, blue cycle and because both together their length was like more than 2K. Either the black cycle has length greater than equal to K or blue cycles length created. So, now you have shown that if G has a cycle of length at least the G minus uv has a cycle of length at least K.

(Refer Slide Time: 11:23)



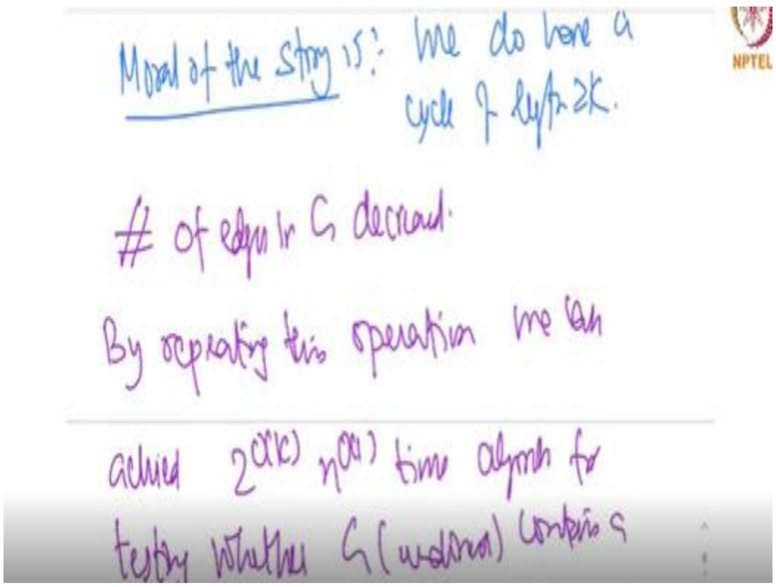
But what more we do. If G contracted uv has a cycle of length greater than K then G has a cycle of length greater than K. Because look at this cycle in G - uv. If uv is not on the cycle then this is also a cycle in G. And now suppose uv is there in the cycle. But then what do you know about uv is an edge and what you know this is a neighbour of K. So, now let us look at this case.

(Refer Slide Time: 12:27)



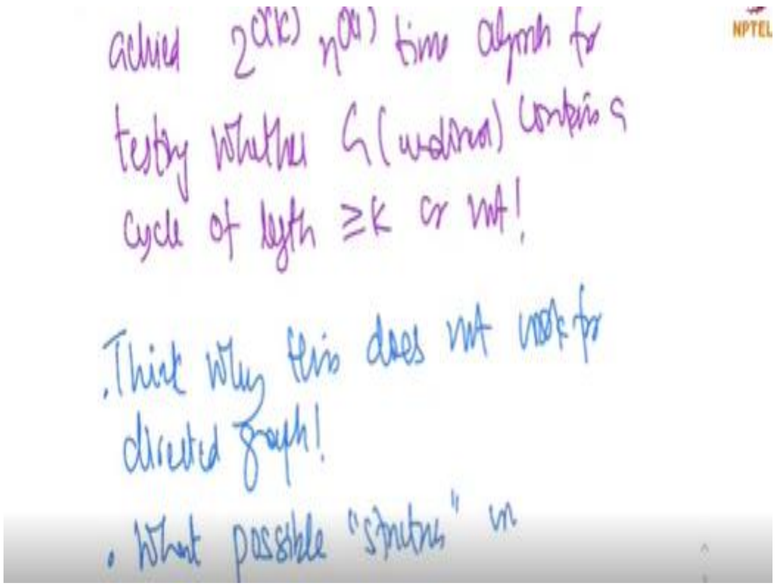
So, look at x y and uv now on this cycle first case x is a neighbour of u, y is also neighbor of u, y is a neighbour of v this is one possibility. x is a neighbour have v, y is a neighbour of u, x is a neighbour of v y is a neighbour of v, the other four possibilities. Now if x is a neighbor of u and y is the neighbour of u replace uv with u and then you get a cycling G in this case. And x in the neighbour of u and y is in the neighbour of v then you just take you take x u, u have this edge and then y, cycle length would not have shrunk. And similarly here.

(Refer Slide Time: 13:49)



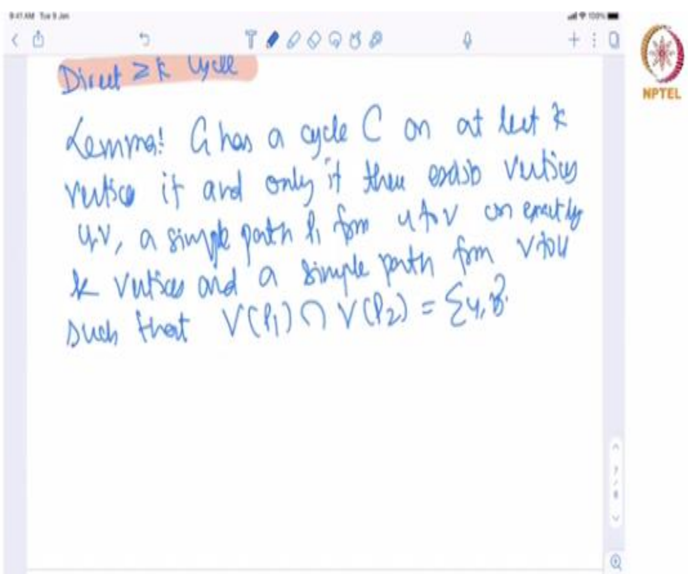
So, the moral of the story is that we do have a cycle of length at least K. What did we gain number of edges in G decreased. So, it is like a one way function. So, by repeating this operation we can achieve 2 to the power big O of K n to the power big O of one time algorithm for testing whether G undirected contains a cycle of length at least K or not. So, that is it. So, this is for the undirected graph.

(Refer Slide Time: 15:22)



But good thing or rather I would say think why this does not work for directed graph. Secondly what possible structure we could find to contract to imitate the above algorithm. So, we took a digression and I told you a little bit slightly interesting elements. This is like more like a puzzle for undirected graph, but it is a puzzle which is worth to do.

(Refer Slide Time: 16:29)

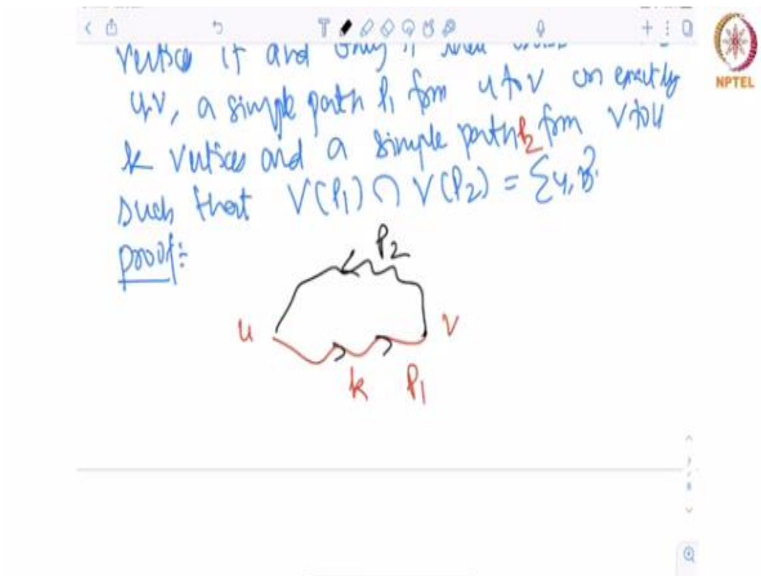


So, now we go back to the setting of directed at least K cyclic is. So, we have gone back to. Now, let us try to so one simple observation of the lemma that will at least get us started and that is like a starting point for some of this algorithm. The basic lemma is G has a cycle C on at least

K vertices if and only if there exist vertices u, v a simple path p 1 from u to v on exactly K vertices and a simple path from v to u such that if you look at the intersection is basically u, v.

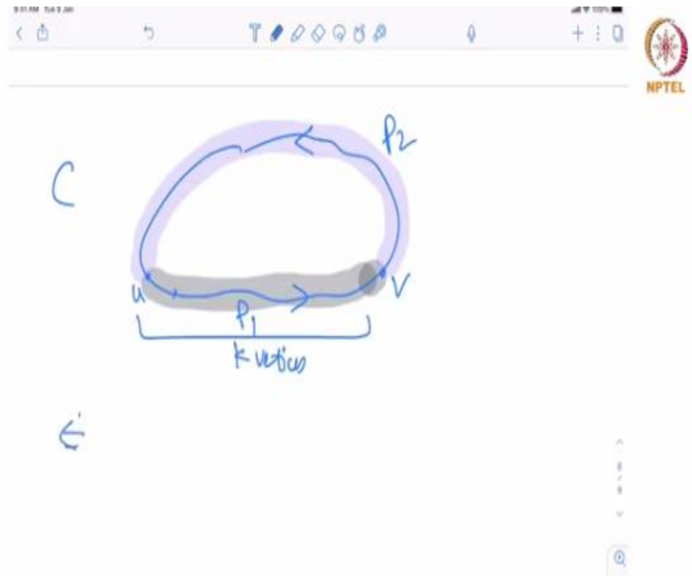
What is this tells us that look the lemma says proof is much simpler and the statement that if you do the cycle on at least K vertices then we can find the following thing. What is the following thing? I can find you if there exists some vertex u and v.

(Refer Slide Time: 18:34)



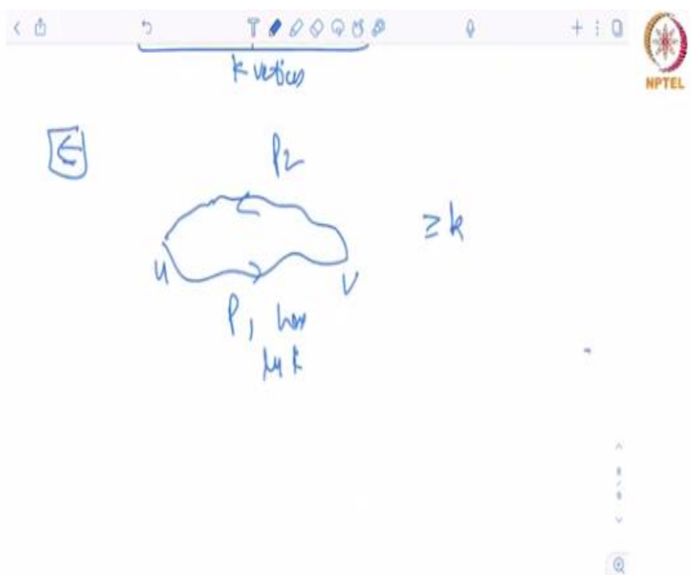
I can find a some vertex u and v and a simple path and this content exactly K vertices. And so, what this is what I call P 1 ? And I can also find a path from v to u and the simple path from v to u call that p 2 from v to u. So let us so there is a simple path from P two from v to u and whatever property of this path the intersection of these two is only at v to u. So, this will imply a directed cycle of length. Why is this true?

(Refer Slide Time: 19:22)



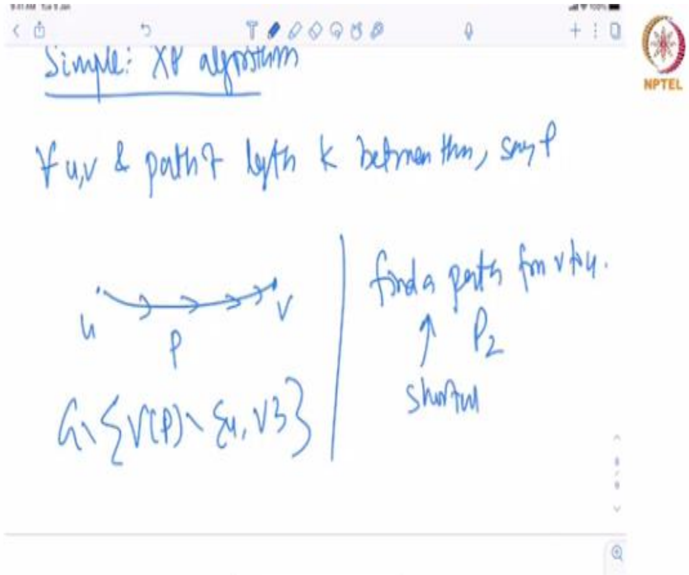
Let us that is the proof is very simple. You pick up a cycle, so this is your C start like pick up the first K vertices, this is exactly first this length is K vertices. You call that P 1 and take this part of the path as P 2, that is it. This is your P 1, this is your P 2 that is it. So, if we have a cycle, we can find such a path. And the backward direction is also obvious.

(Refer Slide Time: 20:19)



You look at a path u to v, this is your P 1 has length k, and there is another path v to u could be just an edge actually. By the way, that could just be an edge, but still you are able to close a cycle and the length of the cycle is at least K. So, this is it. Now, notice that this immediately this lemma tells us a simple experience for them at least. It provides what is that?

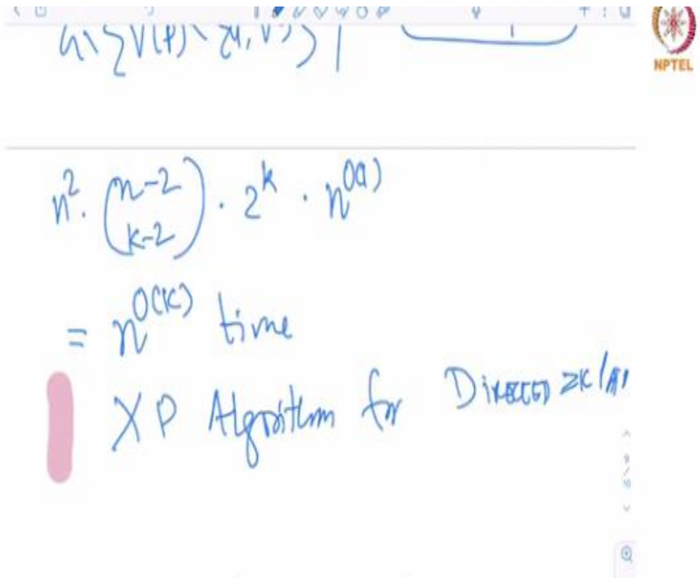
(Refer Slide Time: 21:04)



So, what this tells us, that look basically tells us that for all uv and part of length K between them say p. So, you enumerate all the path have a numerate, it does not matter the path on this. Now, what will you do? You look at G minus internal vertices. So, basically you look at G minus V p all the path, but V p minus u, v. So, basically this is a. Now in this graph find a path from v to u that is it.

Sorry, for me to find. So, now, you know by definition this is a path which disjoint from all the internal vertices your path P and this will act as a P 2. So, for any uv and any P for able to do this, you can concatenate this and just say that there is a cycle of length at least K. And find a path from v to u that is it find a path, I mean I am not saying just find the shortest path for all that you get because this can be done in polytype. So, this step can be done in polytype and how many choices of uv we have?

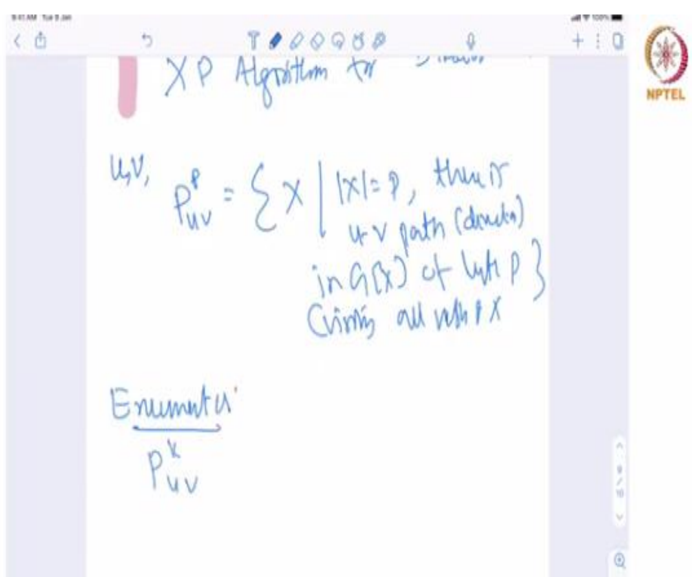
(Refer Slide Time: 22:36)



So, the total number of choices of uv we have is like n squared, how many choices of internal vertices we have is like n - 2, K - 2. And I need to test whether there is a path between them or not. So, that is going to take two to the power K. Check whether this uv and this K -2 vertices. So, this is the running time for this person. And for each of them I need to find P path which is like polynomial time.

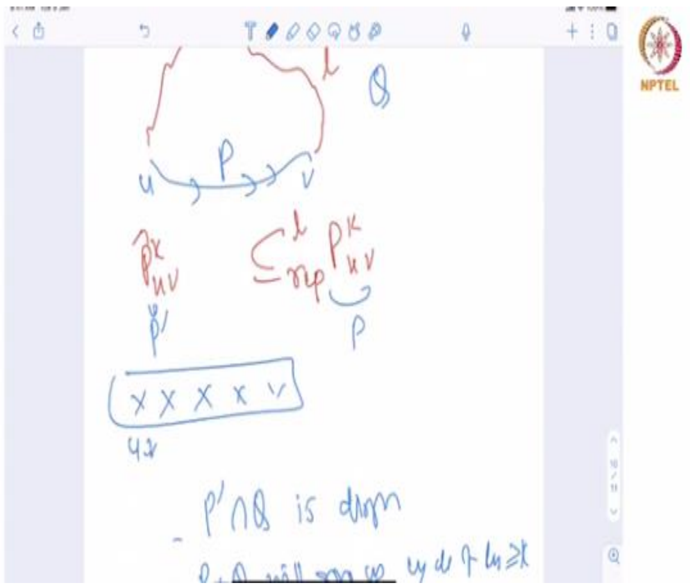
So in total, this is like roughly n to the power O K. So, we are able to design and XP algorithm for directed K path. But this is not what we want to do. So, now let us try to take this algorithm.

(Refer Slide Time: 23:49)



So, our algorithm is very simple. Let us recall, we have u v. So, what is we did, so let us call this P uv K, all the sets X mod X equal to K. Let us rather let us say not K, but like just to that the definition P mod X equal to P there is a uv path directed in graph induced on X of length p. You can say visiting all vertices. So, this is and what we did here, we try to enumerate P K uv.

(Refer Slide Time: 25:15)

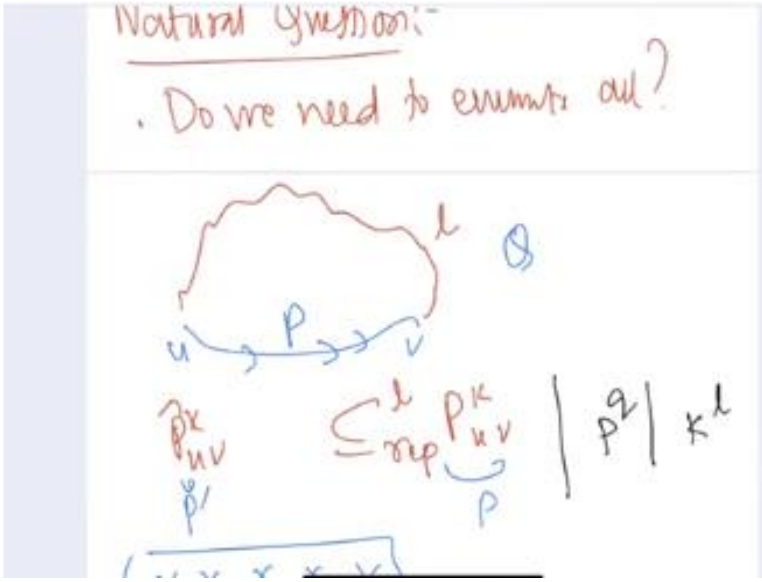


Natural question do we need to enumerate all? So, the point is look at uv. Look at set P K uv. If I knew that this directory cycle has length look at this directed cycle I said look if I knew that this has some length L the red part then what I could do that I could compute the P K uv. I could compute let us say you P hat rather let us be P K uv. I would have computed PK uv hat which is a representative but it is representative with respect to L.

So, what is the meaning of this? I say look so, if other computed this then what would have kept what would have done. So, for every uv, I compute P hat K u v which is an L representative for them. It means, it would have meant that I have kept some family of sets between u and v that look you give me any l length path and that is a set for which I have something which are disjoint.

Then I have something which is disjoint in my set also meaning which will mean that suppose you would have given so, basically what it means that you look at any l and cycle and suppose this is some set q. And I know that suppose this is P. So, P is inside this and P and q are disjoint now. Because of the representativeness we would have found some P prime. So, the P prime intersection q is disjoint and P prime contracted with q will give us cycle of length greater than equal to K.

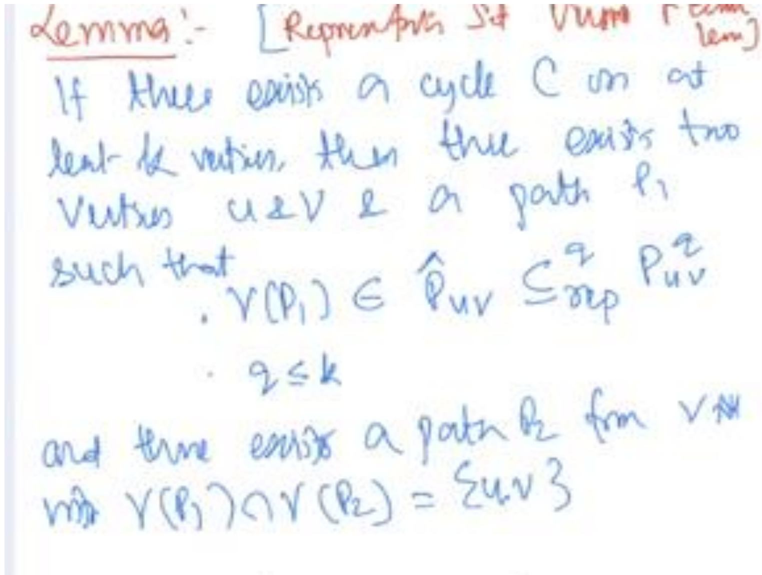
(Refer Slide Time: 28:00)



But the problem is we do not know whether l could be a function of k, l could be much much larger. And if l is going to be much much larger than the computation of this representative set is going to be it depends what are the computation of this representative set. So, the representative set computation if you remember was P plus q or other so this is fine. So, it was P to the power q where q was the future or the set from the future.

So, now P is like k it will be like l. So, this is an algorithm we cannot effort. So, yes, we need not enumerate every path, but it depends on the length l.

(Refer Slide Time: 29:05)

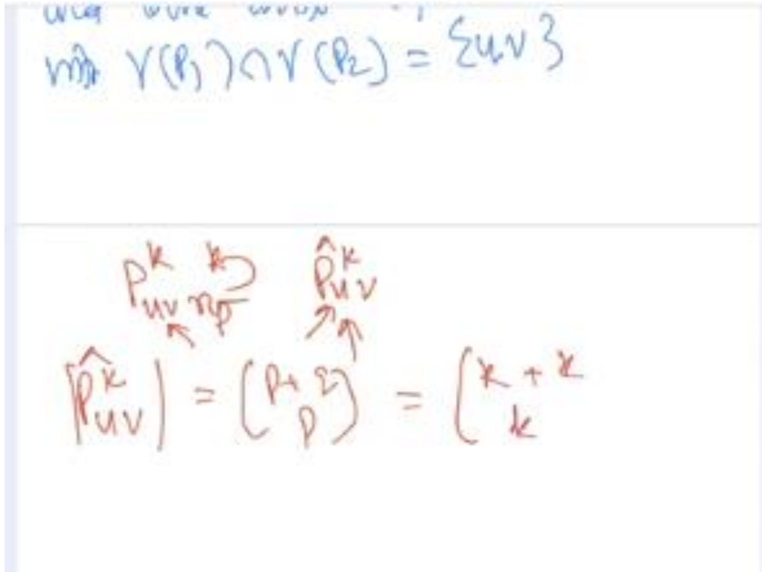


So, this almost worked, but did not work. So, almost worked. But it did not work. So, we will fix it. We will say we will prove l equal to 2k is good enough rather to case in a minute, we will prove that. So, we will prove a lemma, which is analogous to this lemma. But that will be at the heart of our proof. So, our central lemma is this is a combinatorial lemma this is not very hard. But the heart of the proof of this algorithm is the following simple combinatorial lemma.

What is this? If there exists a cycle C on at least k vertices then there exists two vertices before u and v and a path P 1 such that this is important where and a path P 1 such that vertex set of P 1 is an element of it is representative sets is a representative of q representative have P uv q. So, if you look at a path of length k. So, this is a representative of this where what is the value of q? And q is less than equal to k.

And there exists a path P 2 from v to u with vertex set of P 1 intersection vertex set of P 2 is equal to u, v. So, basically it is a truncated version or it is basically you can think of this as a representative set version of the earlier lemma. What I am saying that I will find a vertex u and v says that I will tell you, but I will compute a q representative q is at most k such that like saying that rather than enumerating path from everything I will enumerate paths from my set. And then there exists other paths from v to u which are disjoint from here. So, now suppose q is K.

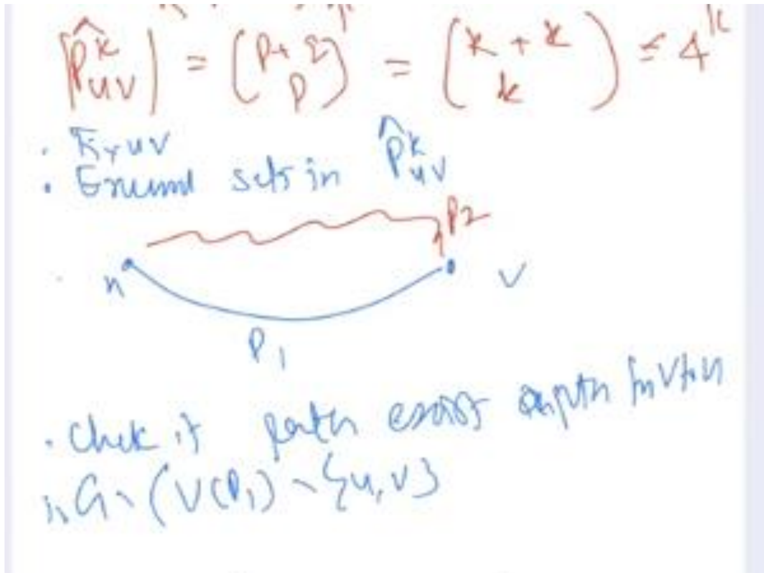
(Refer Slide Time: 33:38)



So, what is the size of this notice? So, every path so, look at P uv k. So, this is a set, I mean and we computed another q representative for this. And what did we computed? We computed P uv k hat and we computed another k representative for this. Suppose we did this then the size of this is going to be how much. P uv k hat it is going to be the size of this is going to be at most 4 to the power k because P plus q choose P which is in our case.

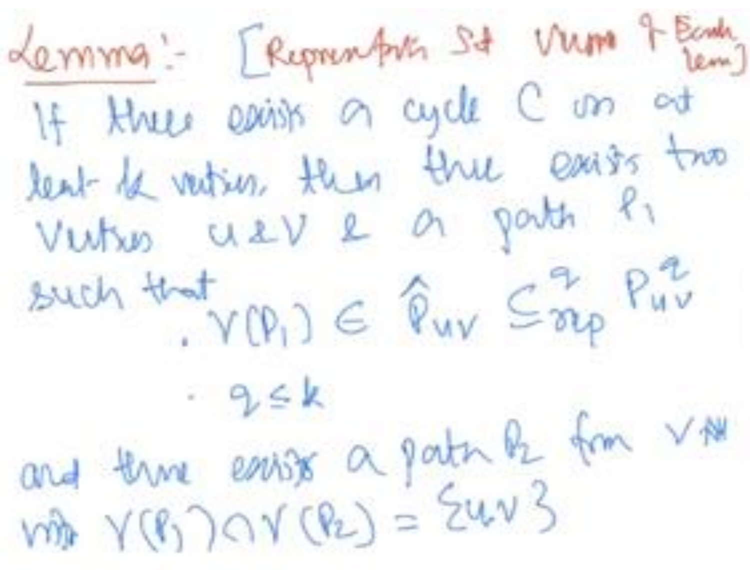
What is P? The lens of paths here. So, the lens of paths here which is P and what is q? The sets of vertices with respect to which I want disjointedness cardinal to the setup set which is again k. So, this is k. So, these 2 k 2 k so this is at most 4 k. So, the algorithm is very simple after this.

(Refer Slide Time: 35:05)



Enumerate sets in fix uv, enumerate sets in P k uv hat, this is P one check if path edges in G minus V P 1 minus u, v just a path from v to u in this. If there is a path that will act as the P 2 and that is a path of length at least k.

(Refer Slide Time: 35:56)

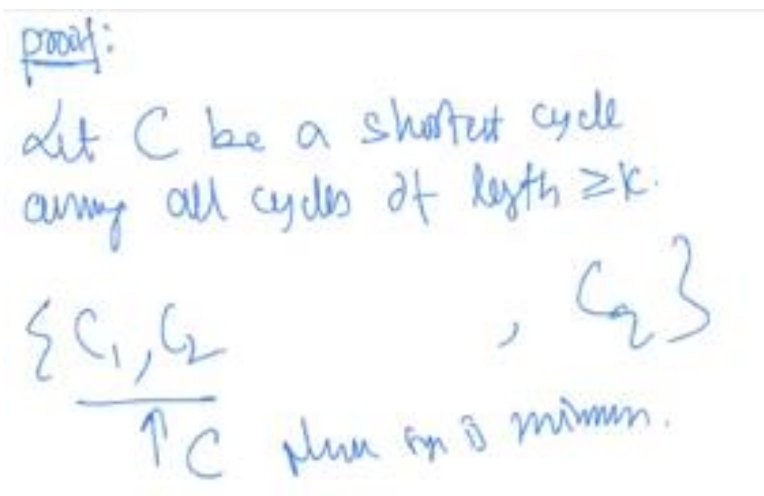


So, the algorithm is the running time of algorithm time to construct P uv k hat, P uv k for all u, v. And so, with our algorithm we know how to compute this by doing the same way we did for K path. You start with one vertex and do all these things and computing K representative. So, you can compute this representative in time k to the power 2k actually because it is a length of k. So, in the beginning like you have two values. When you start with one vertex, you will compute 2k representative.

So, roughly this will take you k to the power big O of K time n to the power big O of 1. So, this is fine you can compute all of them. And now, you have to go through all four to the power k sets in this and run in a polytime. So, the total time is key to the power of big O of k n to the power big O of 1 and if you apply the faster known construction then you had have got 2 to the power big O of k and to the power big O of 1 and there are also that you can optimize.

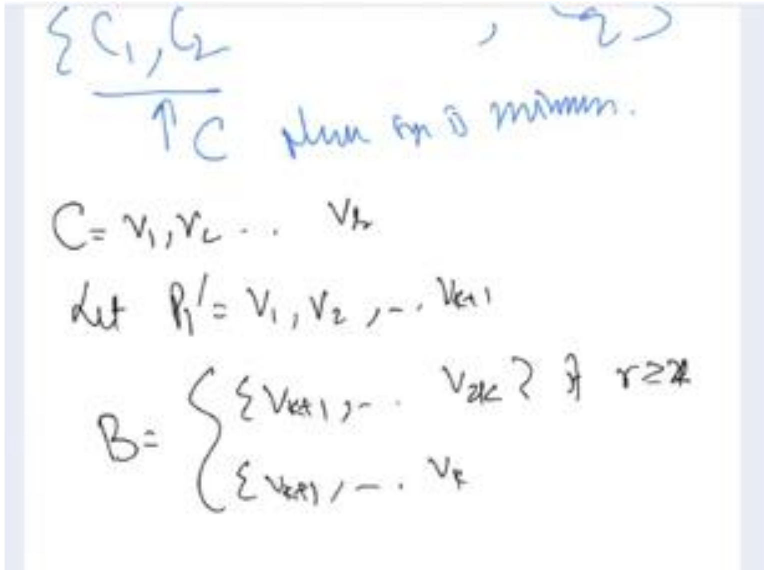
So, if you look at the textbook, it provides you this output in the textbook. So, it all to proving nice combinatorial lemma. So, let us try to prove this combinatorial lemma and once we have proved this algorithm just follows. So, let us just focus on this lemma. Look, so, let us try to prove this lemma, proof is very fairly simple.

(Refer Slide Time: 38:53)



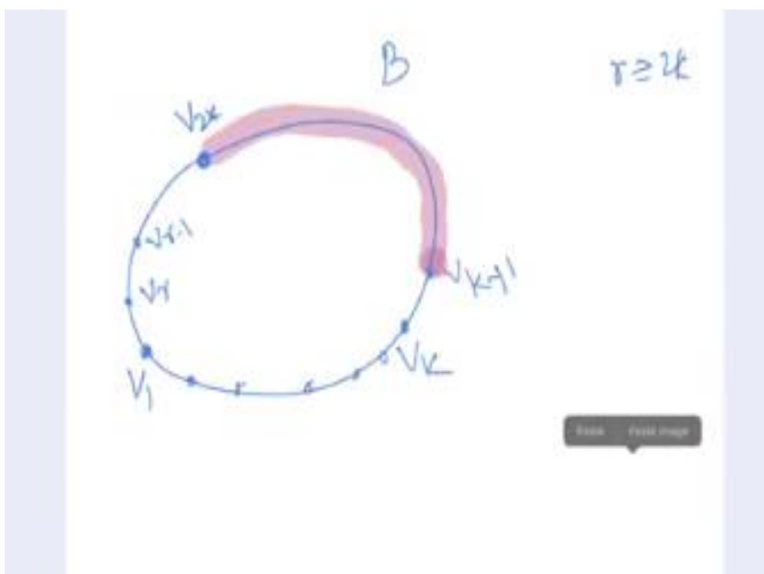
So, prove that you just cute proof, let C be a shortest cycle among all cycles of length at least k. So, there could be many cycles of length at least k. Suppose there C 1, C 2, C q, all the cycles of length at least k. I among them I pick a C whose size is minimum. So, every cycle in this set has length at least k, I pick the one which has a minimum size. There could be many such cycles of minimum size you I pick one. So, this is what I mean. Let us see we are shortest cycle among all cycles of length at least k. So, this is an intuitive proof.

(Refer Slide Time: 40:00)



So, you will just let us say let us write down the vertices of the cycle. So, suppose C is v 1, v 2, V r and now let P 1 prime be v 1, v 2, v k plus 1. And now I define B equal to this is an important is V k + 1, to V 2k. If r is greater than equal to 2k, v k plus 1 V k or V r if r is less than 2k. So, what I did I said look here is my cycle.

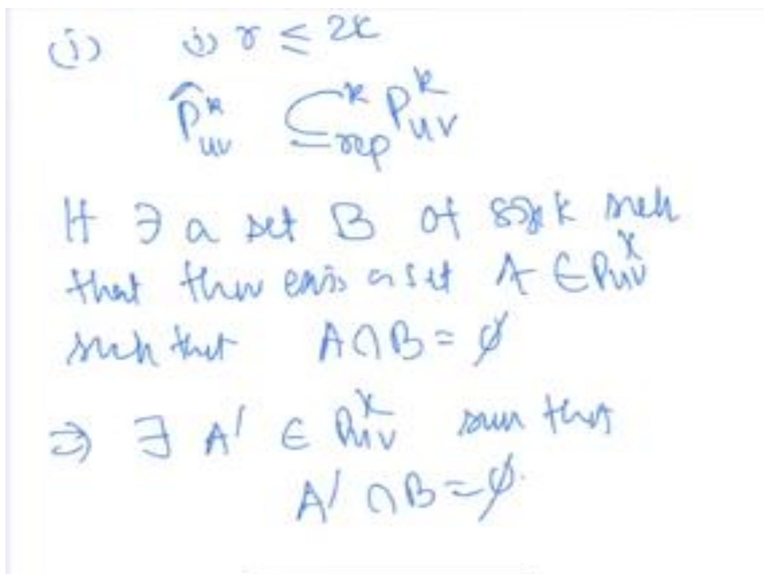
(Refer Slide Time: 41:00)



Here is my cycle. Suppose this is V 1 to V k on this. Now, I look at the length of the cycle, this is what so I said look is in support this is my V r, V r minus 1 V k plus 1. I said look if this cycle is this part of the cycle is here I check. So, I am going to form a set B if r is less than 2k then my B is everything then in that case my B is everything. Otherwise what I am saying to you is it fine. So, this is B when r is strictly less than 2k.

Now, imagine that r is greater than 2k but it could be much more. In that case I start with V k plus 1 and I go and I stop this is my B and what is the property of this, this is nothing but V 2k. So, I do not go further all I go at this point of time if first next k vertices after V k. So, this is how we have constructed our B. Now, we will have two cases. Look at two cases. Look at the first case.

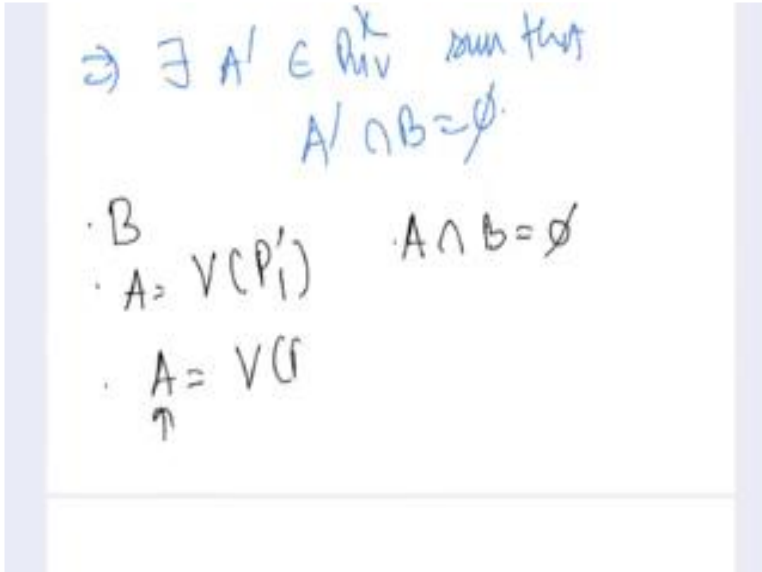
(Refer Slide Time: 43:13)



What is the first case? Suppose r is less than so this is my r is utmost 2k. Now whatever computed now let us look at P v1 vk let us call this v 1 u and let us call this V. So, now let us consider P uv and the representative we have computed up to length K which is P uv k. So, what is the meaning of this? The meaning of this set is that if there exists a set B of size k such that there exists a set A in P uv k;

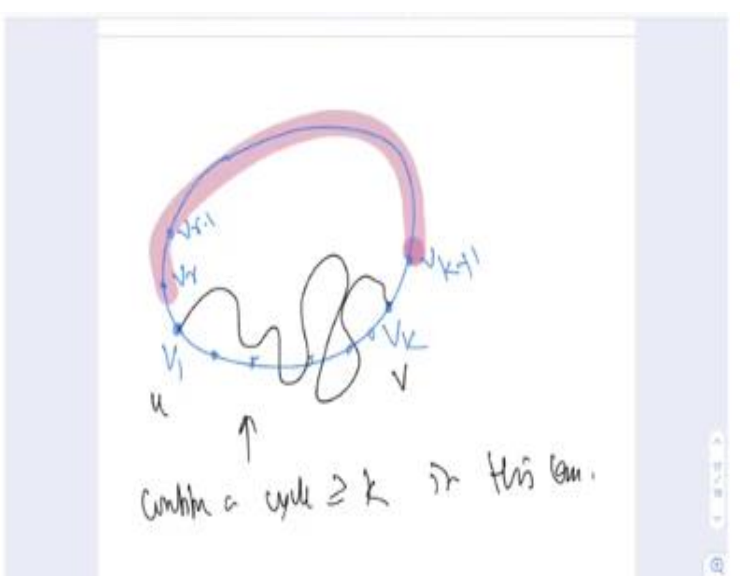
Such that the new set a in P uv k such that A intersection B is empty implies there exists A prime in P uv k such that there is A prime in P uv k such that there exists a prime in P uv k such that A prime intersection B is fine, that is all the definition of this is.

(Refer Slide Time: 45:21)



So, now we are in the case r equal to less than or equal to 2K. So, B is B and what is our A? A is vertex set of P 1. This is P 1 that the path uv. What I know that A intersection B is empty. Then what will you give me? You will give me another path. You will give me a no this is A this is what we call P 1 prime. So, what will you give me? You will give me a set A let us call that vertex set of P 1. And it is intersection of B is phi. So, now I can find a replacement in this case.

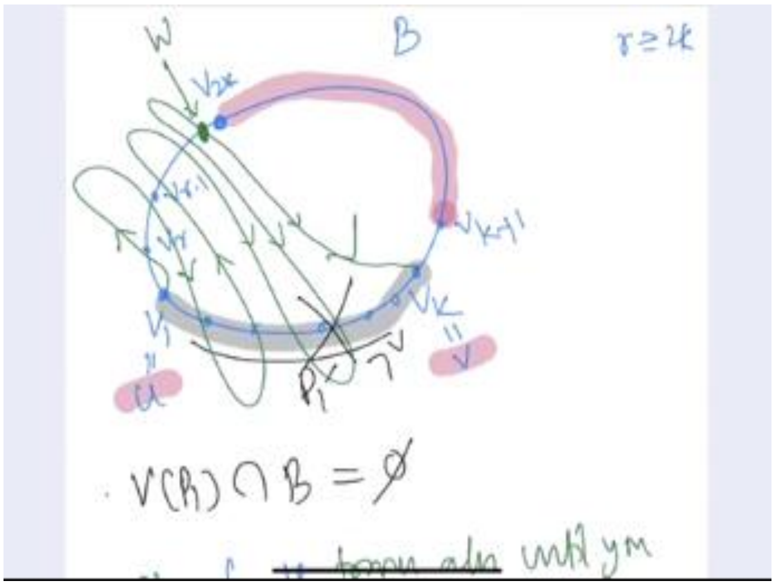
(Refer Slide Time: 46:17)



So, in this case let us so, we are talking about this case. So, now look at this this is what u and this is your v, I can find a replacement. So, now I will find another replacement maybe not this. But like this k, but it is still disjoint from the red part. So, this is going to constitute a cycle of length at least k in this case. So, this is an easy case. So, because the cycle was not too long and you have kept this prefixes that like for every prefixes you have kept a prefix that acts as a disjoint with respect to any future perspective.

So this is perfectly fine. So, the not so easy case is when the second case is slightly more complicated but slightly more interesting. So let us look at this case.

(Refer Slide Time: 48:00)

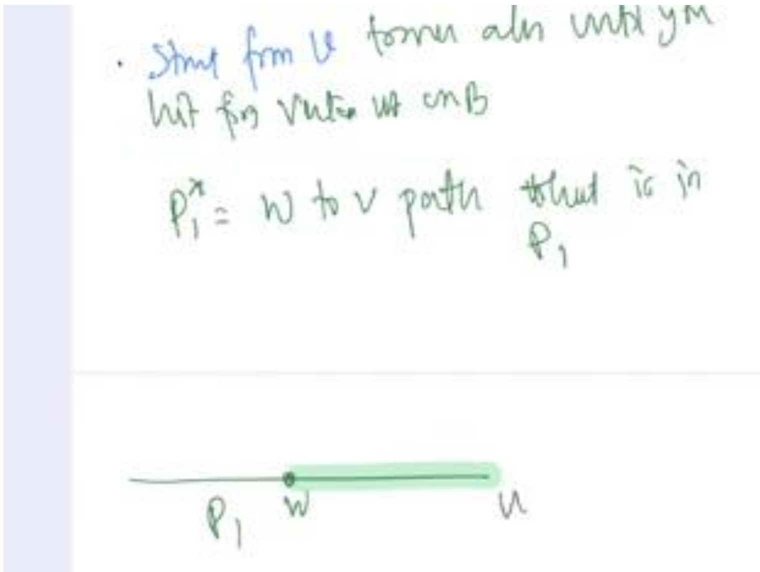


This is the case. Now, what you know? You know for sure that there is a replacement for P 1 prime which is in this case we know there exist. So, if you think of this as your A and this then there exists A prime in P k uv hat which is disjoint from this. Let that A prime we call it vertex set of P. Now let us morph this picture. How does it look like? So, the picture is going to look like slightly different based.

Now all I know that this is the same object. Now, so this I do not know this path exists or not, but I know that some path could exist like this. But what is the property of this path? This path property is that V P 1 is still disjoint from B this is an important point. So, now let me show you. Now, let us start from V and I think this picture is slightly. So, let us say this part we do not know if it exists or not maybe not exist.

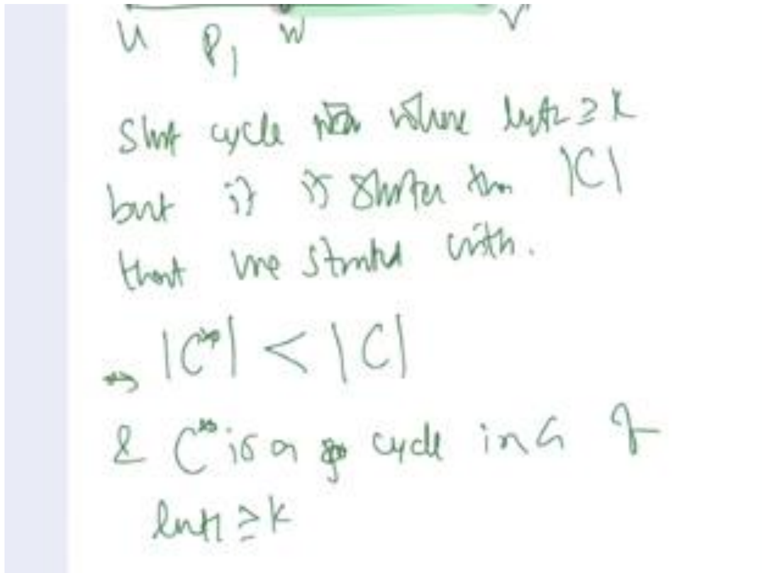
But there exists some other path which starts at u and maybe goes around like this and come back. And the direction of the path is start is this. Now, you start from the v, you traverse along until you hit first vertex not on B. So, suppose you hit up the vertex B, let us call that vertex W. Now so, what is my path? I know that this green path is disjoint from this red path. So, now we can construct the path. How will I construct the path? I will follow the green path.

(Refer Slide Time: 52:01)



So, I am going to have so look at P 1 star as nothing but W to V path which is green that is in P 1. So, basically what happens is this is your P 1 W is somewhere here. So, now we decided to take this path. Now, where does it end it ends at u. And now what do I have another path from u to w which is of this.

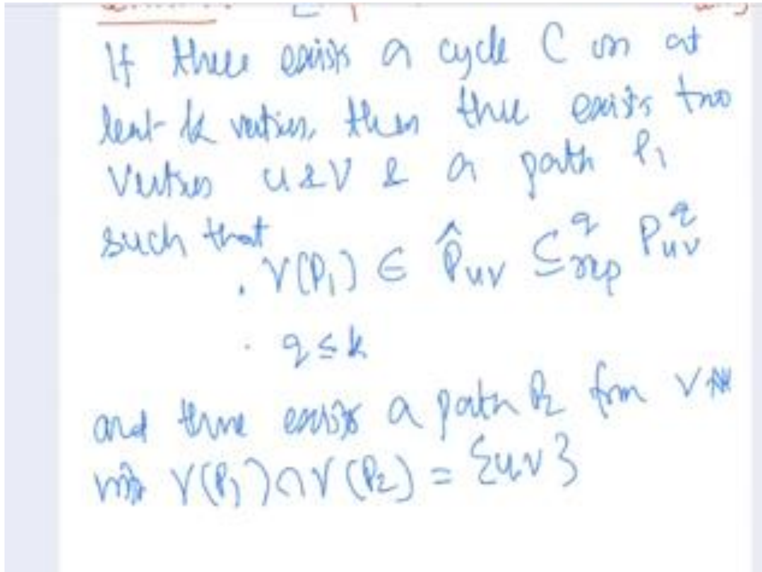
(Refer Slide Time: 52:54)



And starting from you, this is the first vertex that I hit. So, none of these vertices contains any vertex from P 1. What you know, this is length, at least k. So, what I have been able to show, so, we have been able to show that k, k is a short cycle whose length is greater than equal to k, but it is shorter then what that we started with, more see that we started with. So, we can show because look, it definitely does not contain u like it does not contain this should be v starting from the right.

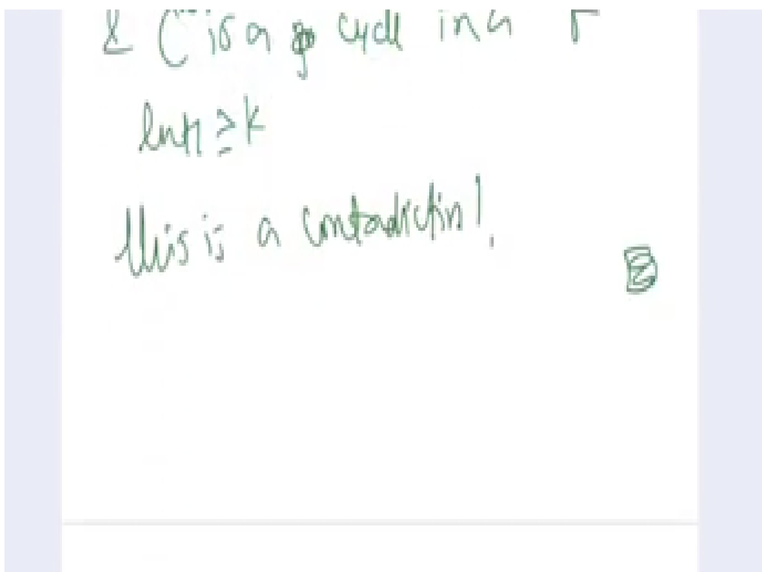
So, the path length of P one was fixed. So, each of these paths are like sub paths of their corresponding thing. So, call this a C star. What a property of C star? C star is strictly less than say you can check for yourself and C star is a cycle in G of length at least k.

(Refer Slide Time: 54:41)



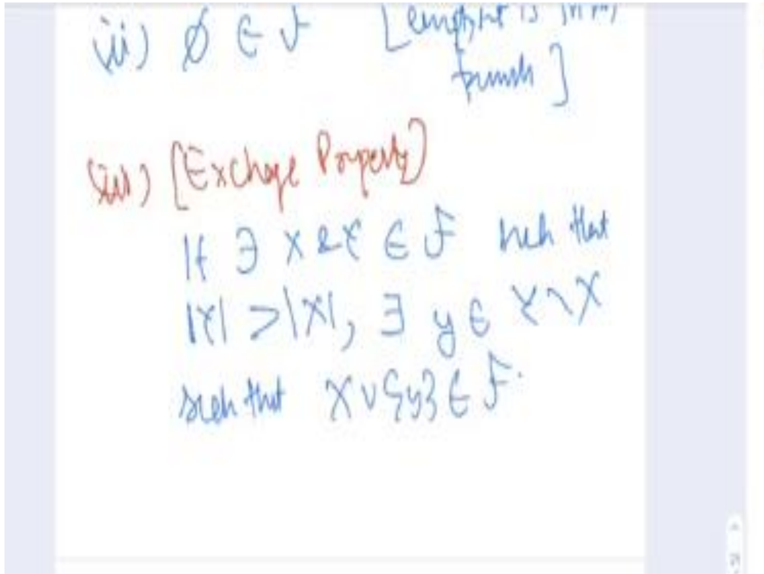
So, this is a contradiction. So, what do we know? So, we know now that our statement of the lemma holds. And in particular the shortest cycle of length at least k has the property that you can find vertices u and v and a path P 1 in my rep set says that there is a path from v to u which is disjoint just fine anyway, so, that is it.

(Refer Slide Time: 55:09)



So, this should end our lemma as well as the proof. And what we have shown is that directed k cycle is FPT. In fact, we give k to the power of k, n to the power of the one time algorithm and that is much faster C power k into the power of one time algorithm.

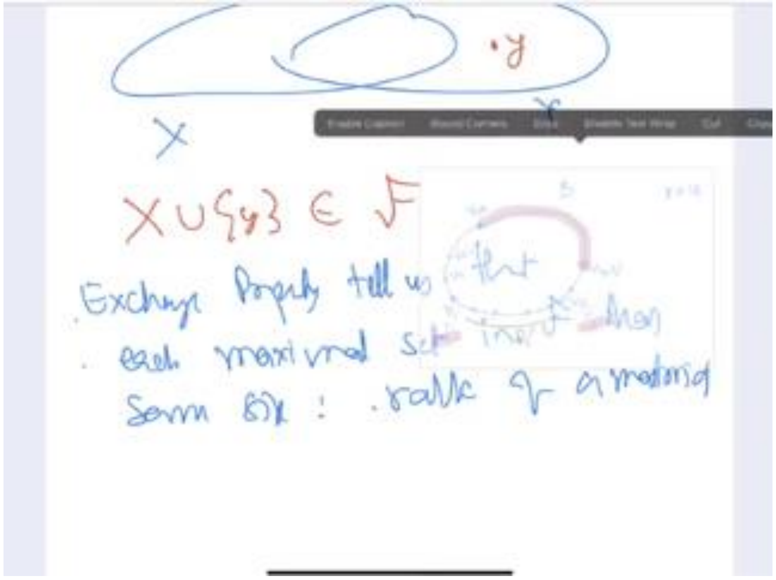
(Refer Slide Time: 55:51)



So, now let me take another 10 minutes and let me talk about matroids. So, up until now, we have talked about set system. But there is another object of our interest or what is called matroids. So, what is a matroids? Matroids is basically it is an universe in the family of sets system. What is the property of this? So, the first property is it is hereditary. If X belongs to family then all subsets of X belongs to family and this is called hereditary property.

Secondly, phi is in my family like that an empty set is in my family and thirdly which is the most important property of matroid is what is called exchange property. And what is an exchange property? If there exists X and Y in our family such that mod Y is strictly greater than mod X that exists Y in Y minus X such that X union Y is our family.

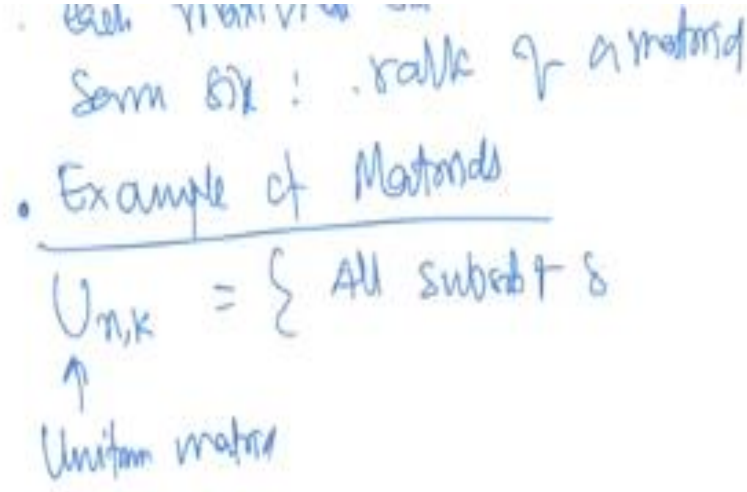
(Refer Slide Time: 57:41)



So, this is what it says that look I have a big set why and I have another set X currently of Y is bigger than X then I can find an element. This is not a picture right picture. So, this is a set X and this is set Y and Y is larger than set X then I will be able to find an element Y here such that X union Y is in my family. So, we can enlarge X by an element that is not present in X but only present in Y to my family.

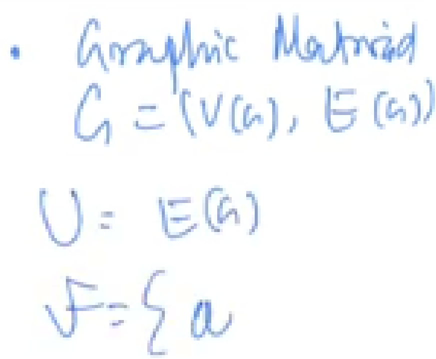
So, these kinds of set families are called matroids. Now notice exchange property tells us that each maximal set in family has same size and these are called rank of a matroid and of course they have a relation to leaner property.

(Refer Slide Time: 59:17)



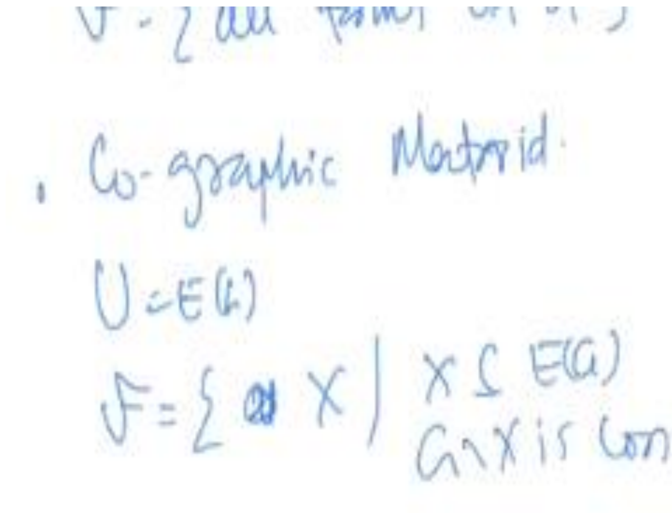
So, this is one thing. Another property which is very right let us look at some example of matroids. So, some examples of matroids, one is something called U n k or uniform matroid which is basically all subsets of size at most k you can take this from symmetric. But there are some very non trivial matroid.

(Refer Slide Time: 01:00:05)



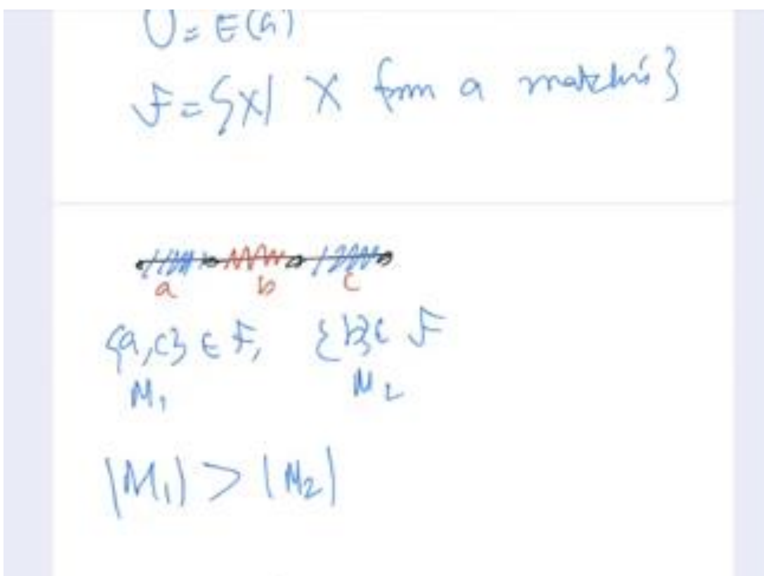
But let us say another one I will tell you from graphic matroid, photographic metric look at a graph G. And this is VG and EG so your universe it is set of G and your family is all forests of G show that this forms a matroid.

(Refer Slide Time: 01:00:34)



There is something called co graphic matroid. What is this again? My universe is a set of my graph and my family is or X, X is a subset of EEG. And G minus X is connected like it is basically dots subset of areas whose duration keeps the graph connected then these are called co graphic Metroid. So, there are various Metroid.

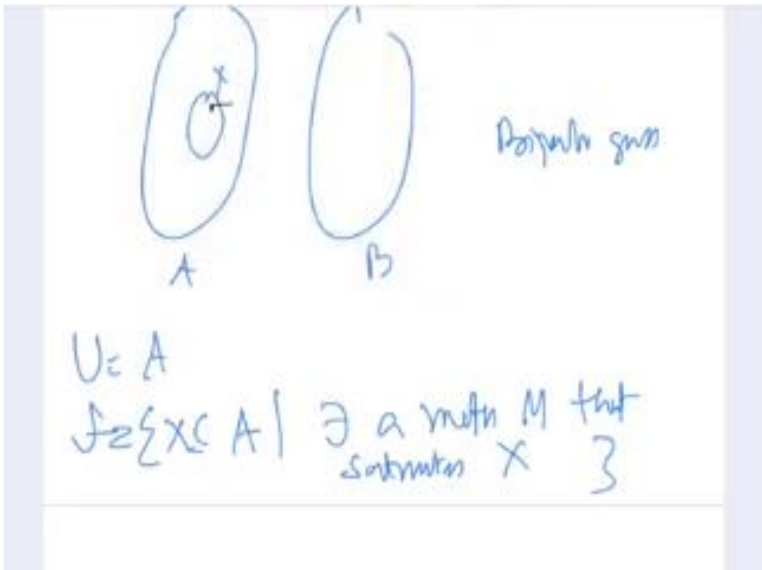
(Refer Slide Time: 01:01:09)



What about matching? So, let us see, my universe is again ad set and my family is X, X forms are matching. So, you can show that if something forms a matching then all their subsets also forms a matching. But you can show that this is not this is not a matroid because look at this example. In my family I will have this set so let us look at this what happens. So, now, there is a let us say there is a red, so blue and red matching. Let us call this says a, b and c.

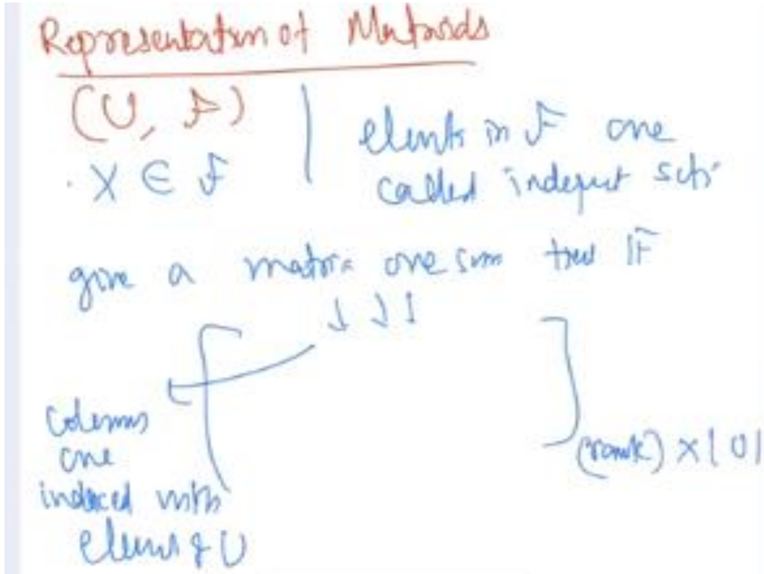
Now, a and c belongs to my family, b belongs to the family. Now, let us call this matching M 1 on this call it definitely cardinality of M 1 is more than cardinality of M 2. But there is no way I can add an edge in either a or c to b because that will not form a matching.

(Refer Slide Time: 01:02:52)



So, matching do not form a matroid. And but there are ways to navigate it. So, in bipartite graph you can have say set A and B bipartite graph. So, you can come up with my universe as A and your family is X subset of X. There exists a matching M that saturates X meaning you have an X here and you have a matching like this saturates X.

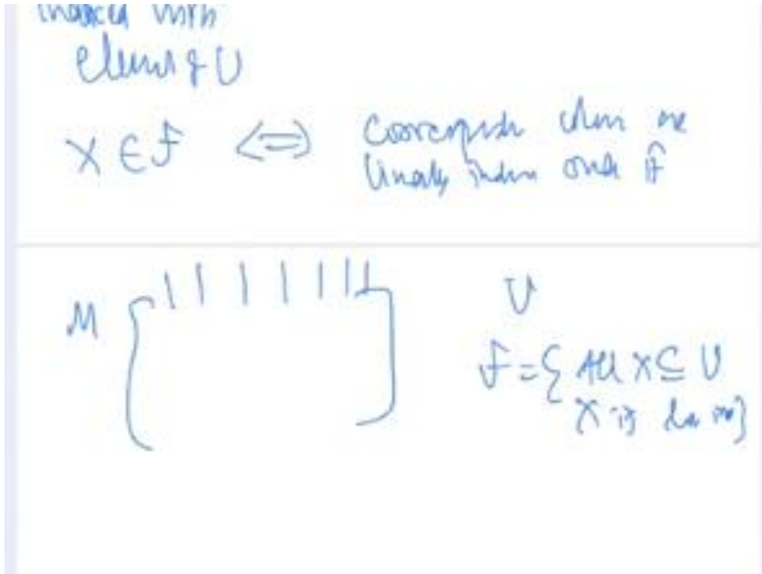
(Refer Slide Time: 01:03:41)



You can show that this forms a matroid and this is called transversal matroid. Another important property of matroid that is useful for us which we like to know is the following something called representation of a matroid. What is the representation of a matroid? Notice that you are given a u and a family like if could be very, very large. So, how do you check whether some X belongs to F or not ? And elements in F are called independent sets.

How do you check whether excellent independent or not? So, of course it could be given explicitly. But the best one of the best way to give this it basically is to give a matrix over some field F. So, you are given a matrix where columns are indexed with elements of u. And generally this matrix you can actually take this matrix size is basically rank times u and what is the property? Look at.

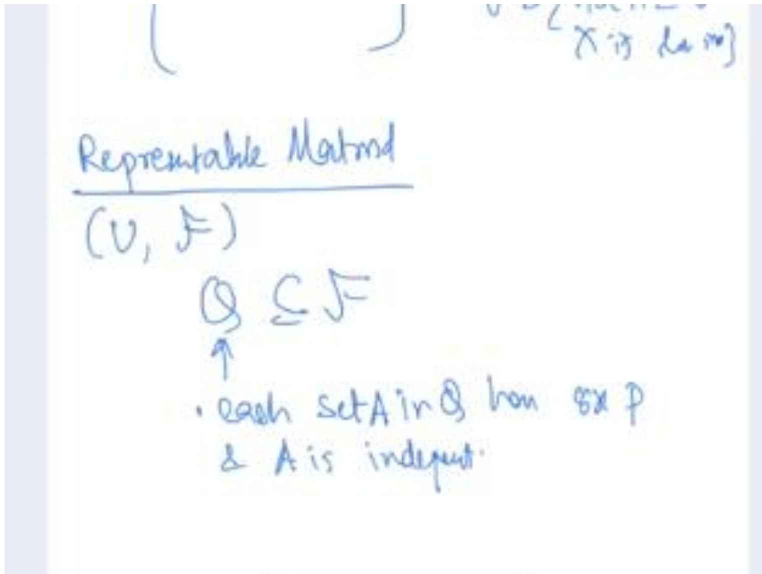
(Refer Slide Time: 01:05:35)



So, now, this is like if you look at X is an f if and only if corresponding columns are linearly independent over F. So, basically what is this matrix tells us this matrix tells us that look, if I wanted to test whether my set X is independent or not, I go and look at the corresponding columns and check whether they are linearly independent are not over field F. So, this is so, a matrix a matroid for which such a matrix exists are called represent table matroids.

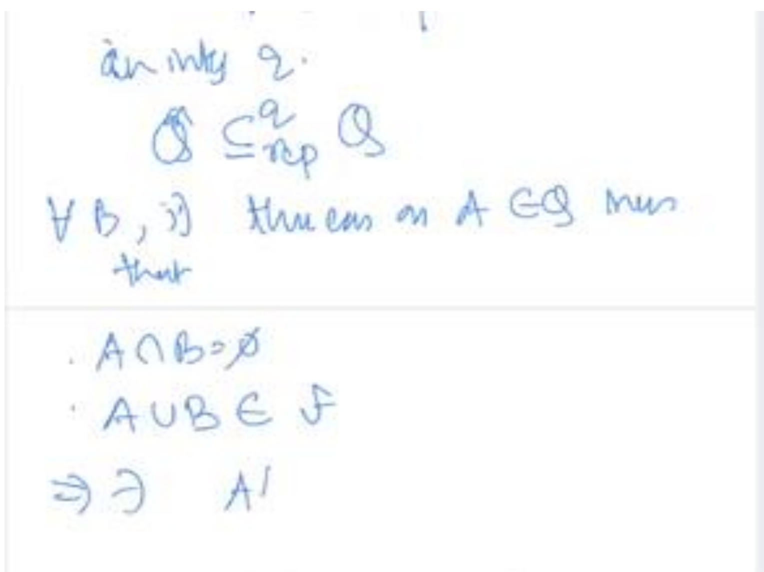
But for that matter if you take a matrix itself and take columns as your universe and what is your family all X subset of u, X is linearly independent then this automatically forms a matrix. And this is why the terms like independent sets are the rank is used in the linear in the matroid. Because it appears that the definition of matroid came from some of these matrix applications.

(Refer Slide Time: 01:06:55)



So, why did they talk about all this? So, there is a theorem about representable matroid. What does it representable matroid theorem is the following. So, you can also define a notion of Q representative. So, I have given u and F this is a matroid and now imagine that u are given some family F, some family let us say Q subset of F. And what are the property of Q? Each set in Q has size P and n and that is more important and each set in Q has size P and A if independent, A is independent.

(Refer Slide Time: 01:08:16)



Because look so, that is independent in the sense that it is or rather fine. Then given an integer q we can come up with Q hat which is a Q representative of Q. And what is the property of this?

Well for all X say or other for all B if that exists and A in Q such that intersection B is phi first and A union B if in our family implies there exist B there exists in A prime.

(Refer Slide Time: 01:09:22)

- $A \cap B = \emptyset$
- $A \cup B \in \mathcal{F}$

$\Rightarrow \exists\ A' \in \hat{Q}$

- $A' \cap B = \emptyset$
- $A' \cup B \in \mathcal{F}$

- given Q, q we can compute $\hat{Q}$ in time $f(p,q) \cdot n^{O(1)}$

There existence A prime in Q hat with a property that a prime intersection B is phi. And more importantly so, in some sense, this rep sets not only preserve disjoints it also preserves independence. And what is known is that given Q and integer Q, we can compute Q hat in time F p q n to the power B. So, it is like in this time, it is quite actually, it is quite, I will maybe I should tell you the running time itself. So, let me tell you the running time it is quite efficient.

(Refer Slide Time: 01:10:24)

- $A' \cup B \in \mathcal{F}$

- given Q, q we can compute $\hat{Q}$ in time $f(p,q) \cdot n^{O(1)}$

$O\left(|Q| \binom{p+q}{p} p^{\omega} + \binom{p+q}{p}^{\omega-1}\right)$

operation over GF(2).

$|\hat{Q}| = \binom{p+q}{p}$ [Lovasz 1970s]

So, you can actually compute in time big O of size of Q p plus q plus p p to the power omega where omega is the matrix multiplication p plus q choose P omega minus 1 operations over some GFB. So, look, I am just giving a overall idea. I am not trying to do everything. And what will the size of Q hat? Size of Q hat is again going to be p plus q choose p. And this is basically a isomerization of a lemma proved by Louvers generalization was lemma proved by Louvers in 1970s.

So, basically all I am trying to tell you that all the all the things about representative sets can be extended to matroids algorithm table matroids. And they have found also lots and lots of application in designing parameterize algorithms. But somehow they are slightly more complicated. And in my opinion, that is not the right place to teach your fit in the first course in parameterize algorithms.

So, with that, let me take an end and this was also my last class in this course. So, thank you for listening. And the next class will be taught by Neil and that class will be about techniques or tools to prove hardness in parameterize algorithms. So, have a good time. Thank you.