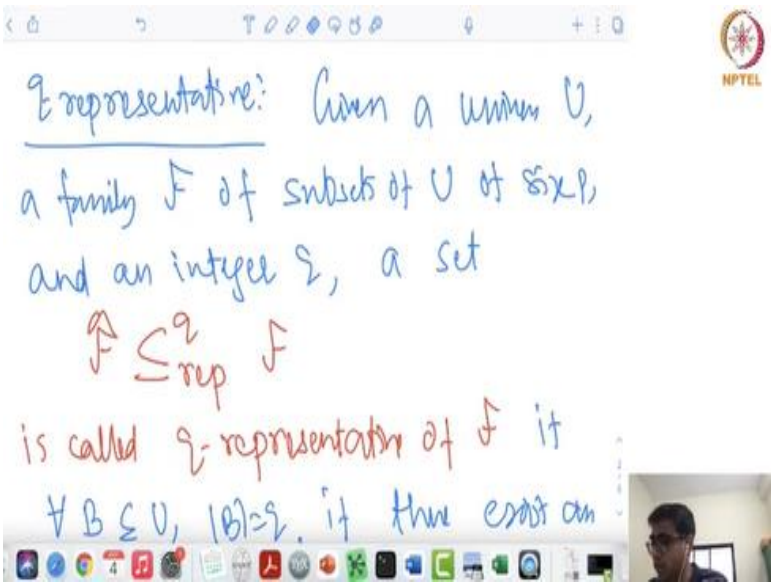
Parameterized Algorithms
Prof. Neeldhara Misra
Prof. Saket Saurabh
The Institute of Mathematical Science
Indian Institute of Technology, Gandhinagar

Lecture – 45
Matroids: Representative Sets – Applications (Paths and Kernels)

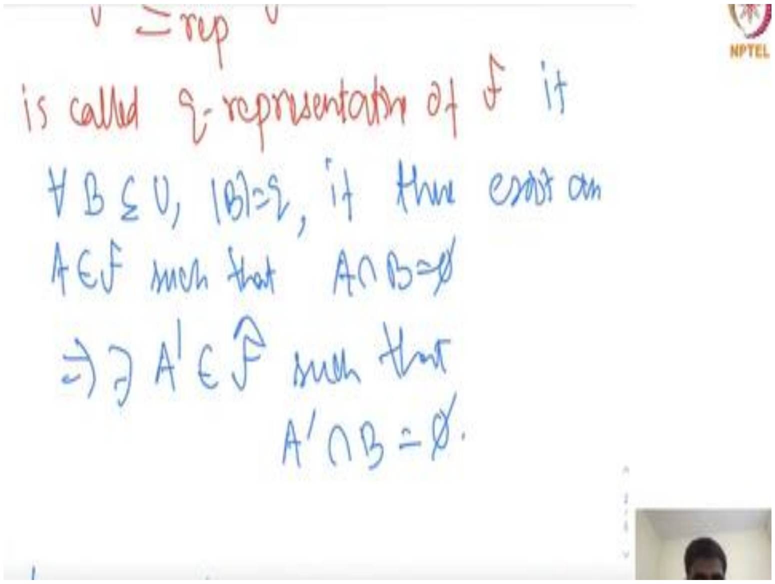
So, welcome to the third lecture on representative sets. And this will be about applications and let us see how many applications we are able to do in today's lecture.

(Refer Slide Time: 00:26)



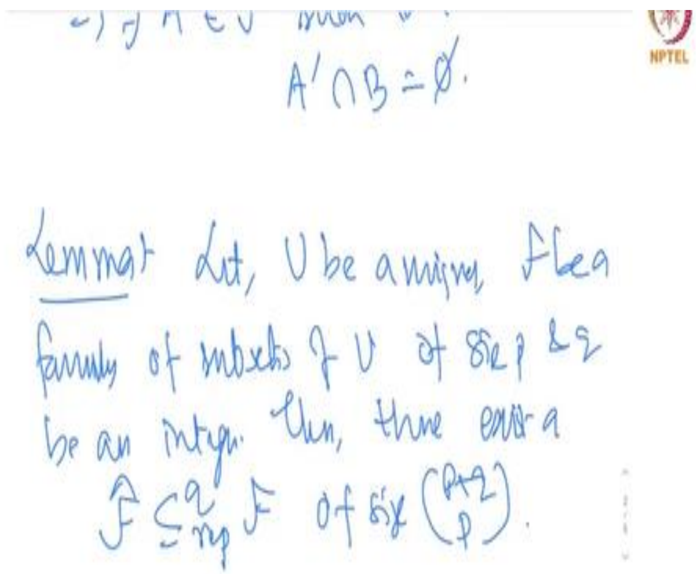
So, first let us just recall the definition of q representative. You are given a universe U, a family F of subsets of U of size P, and you are given an integer q, then you can compute a subset of F, which we call F hat and this is called q representative. Because given any set B of U, whose size is q, if F hat a set which is disjoint from B, then F hat also has a set which is disjoint from B.

(Refer Slide Time: 00:51)



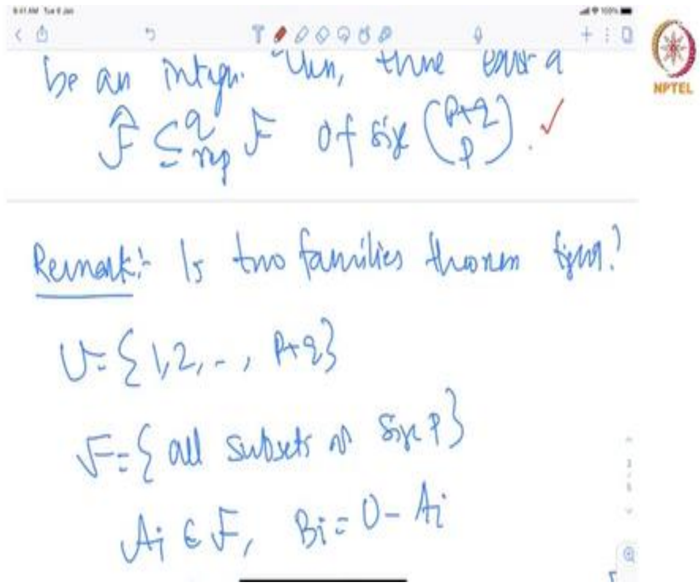
So, F hat satisfies whatever disjoint is property with respect to sets, it could satisfy this will also satisfy.

(Refer Slide Time: 01:06)



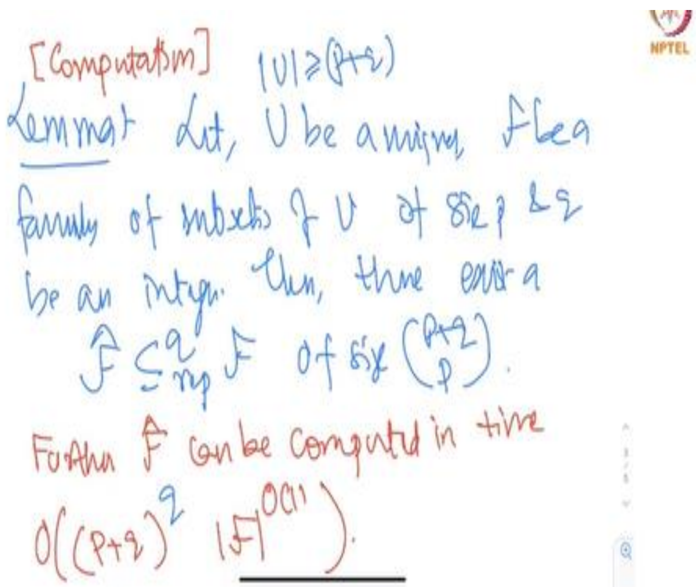
And, then we said look given U are universe given such a set F, in fact there is an F hat whose size only depends on the representativeness, that you want and the size of sets that are in F, and that is of p + q choose P.

(Refer Slide Time: 01:27)



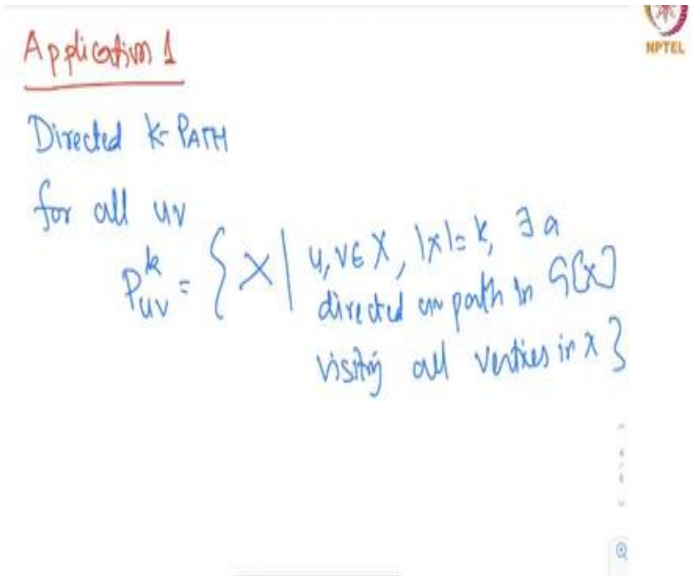
And then we also saw that this is tight.

(Refer Slide Time: 01:29)



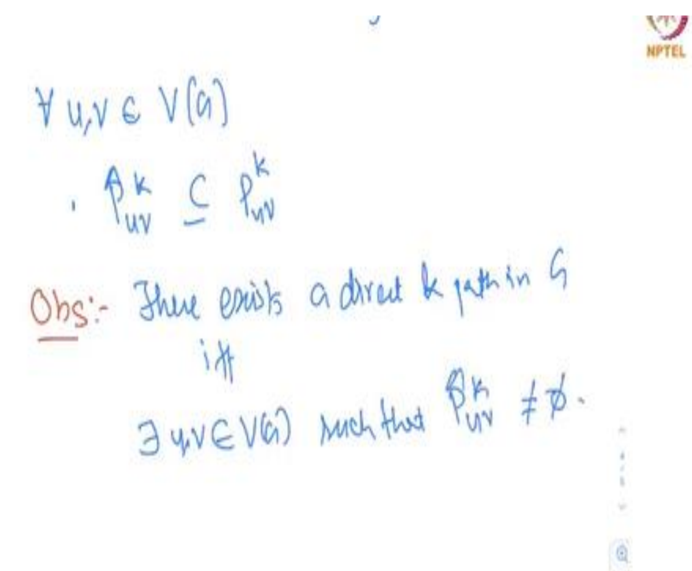
And then we finally saw our computation when universe size is greater than equal to p + q, then actually you can compute an F hat of size p + q choose P, in time p + q to the power q mod F to the power big O of one.

(Refer Slide Time: 01:47)



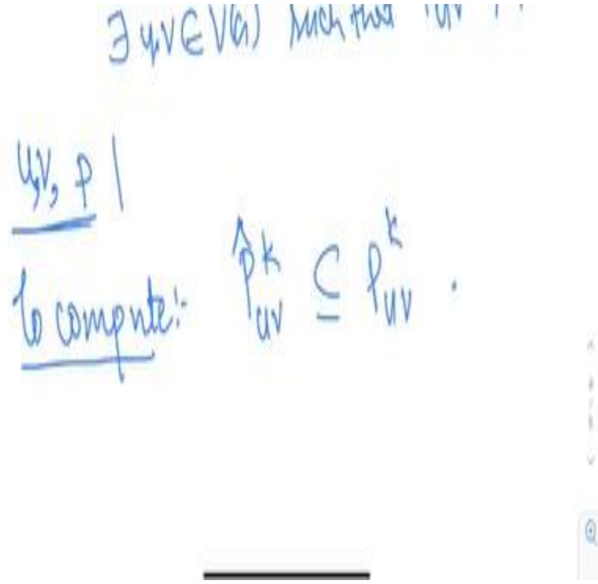
So, now we would like to so its application for computing. Let us see the first application of this, so the first application will be the example that we were trying to build. That is directed k path. So, what are we going to do? So, for all u v, let us P u v k is equal to set X, u come up in X mod, x = k, that exists a path, there exist the directed u v path in graph induced on X visiting all vertices, so this is of length.

(Refer Slide Time: 03:01)



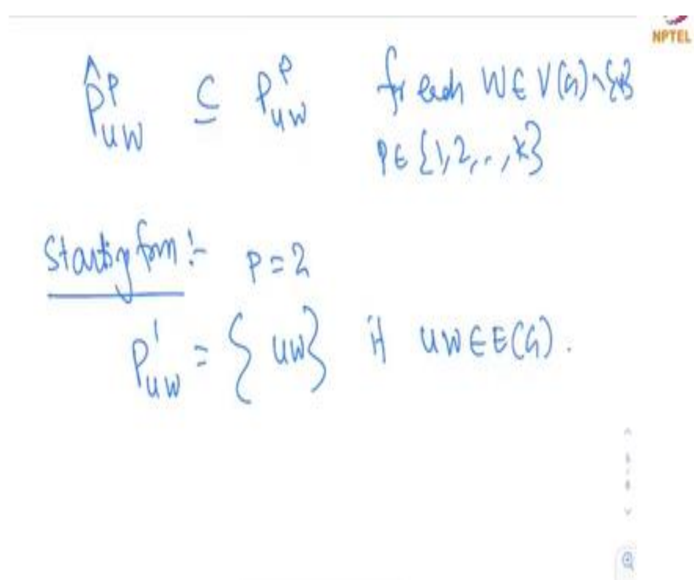
So, now what are we going to compute for all u v in v of G, we are going to compute a subset of P u v k. And notice once we have, so the very simple observation is that exists a directed k path in G, if and only if there exists u, v in V G such that P hat u v, k is not equal to empty. So, this is an observation that we will try to.

(Refer Slide Time: 03:55)



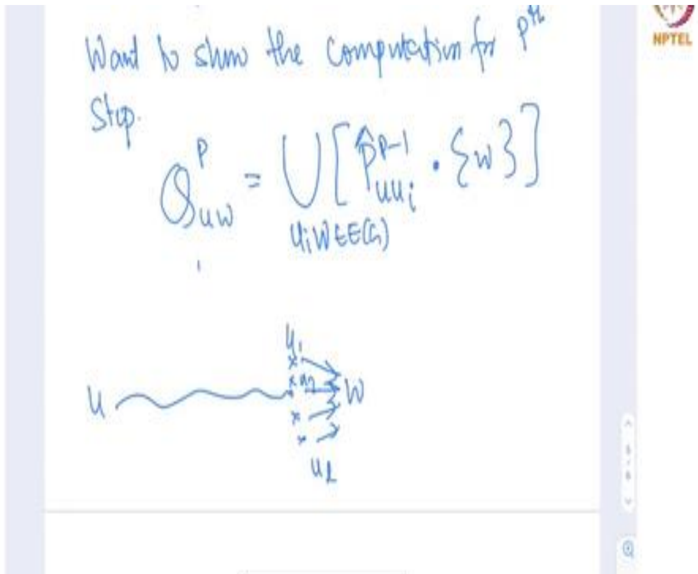
Now, how are we going to compute, so let us fix one u v and some integer P. Now, we would like to compute. So, first what is our objective this is given this to compute, what is our objective to compute P hat u v k, which is contained inside. So, this is our objective. Now, let us see how we are going to compute this.

(Refer Slide Time: 04:50)



So, we are going to compute actually, we are going to compute P u w or rather let us say, we are going to compute P u w P hat of P u w P, P sub integer, for each w in VG - u. And PE it is some integer between 1, 2, k. So, this is what we are going to compute. So, we are going to compute this starting from P = 1. So, P u w 1 is what? So, since we are talking about vertices, let us say, so this is equal to basically u, rather let us say P = 2, we can start from P = 2. So, now P u w is what? It is u w is not H, so this is what.

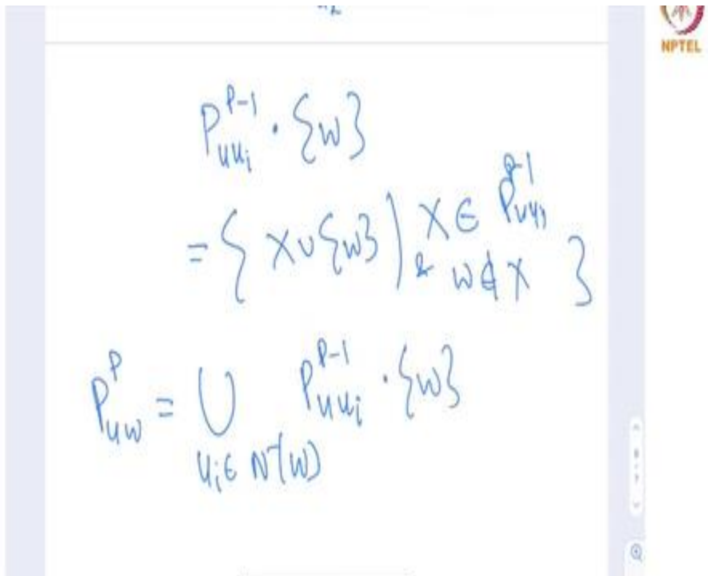
(Refer Slide Time: 06:51)



Now, suppose we want to compute, so this is what it is. Now, suppose inductively we have computed, inductively we have computed P u z P-1 hat of P. So now want to show the computation for Pth step. So, suppose I am going to first create a set Q u w which is going to be. So, now what is this and length P? So, basically think of this is a look at a path, directed path of length u to w.

So, how can you reach? So, you must reach one of its in neighbours, let us say some u 1, u 2, u l suppose so, what I am going to do union over. So, we are going to say, that I am going to keep P hat now, u u i P - 1, u i w is in EE G. And I will tell you what this symbol will mean? So, what I am going to say that look at. So, basically if I were trying to create paths to w.

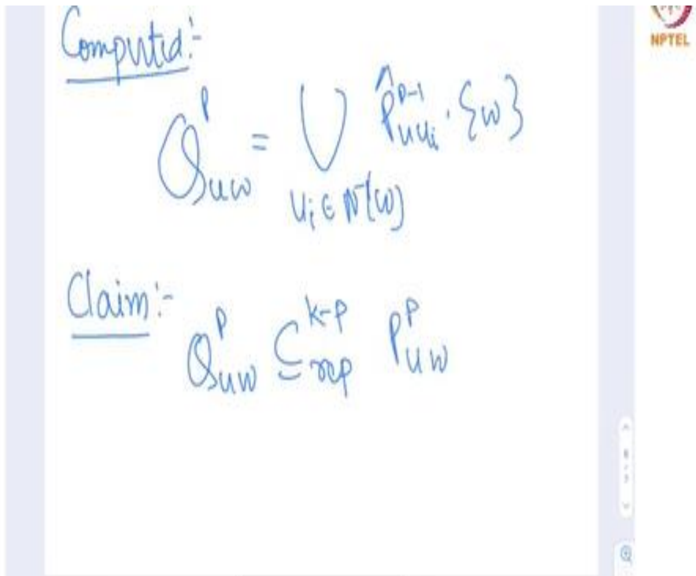
(Refer Slide Time: 09:10)



So, that will consist of P - 1 length paths to you to 1 of these u i, it is in neighbour and then a direct edge from u i to w. So, what I am going to say that look at fix some u i, P hat. So let us look at P u u i, P - 1, this is what it is. Now, so I have this set given this set I am going to create a set w. What is this set w I am going to create? It is going to be X union w, X in P u u i P - 1 and w does not belong to X.

So, w is already not in the set x because otherwise its length will become small. So, this is exactly what it means by this. So, what have you done? I said look rather than creating P pi -1 w. So, notice that what is P u w P is basically if you look at this set, it is union of u i in neighbourhood of w, P u u i P -1 dot w, this operation. So, this is what P u w P is. So, these are those P length path starting from u ending it w, these are the sets. Now, but I did not compute this, what did I compute?

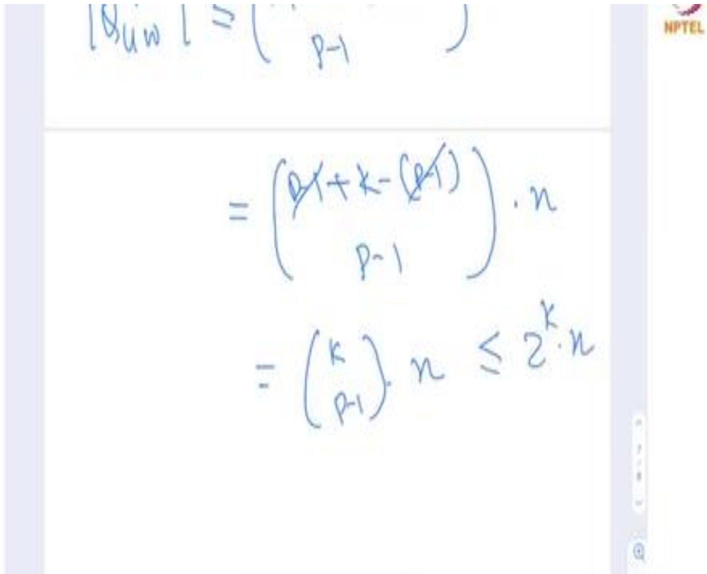
(Refer Slide Time: 10:51)



We computed rather a new set or did a compute, I have computed Q u w P, which was a set rather than I am still going over union u i N - w, but rather than taking dot product with P, we have computed the rep set. But, what kind of, so we will inductively assume that, this is a rep set of size K - P - 1. And I have compute, so this is what I want to say. So, computed, so this is K - P - 1 size, because in the future K - P - 1 vertices are supposed to come.

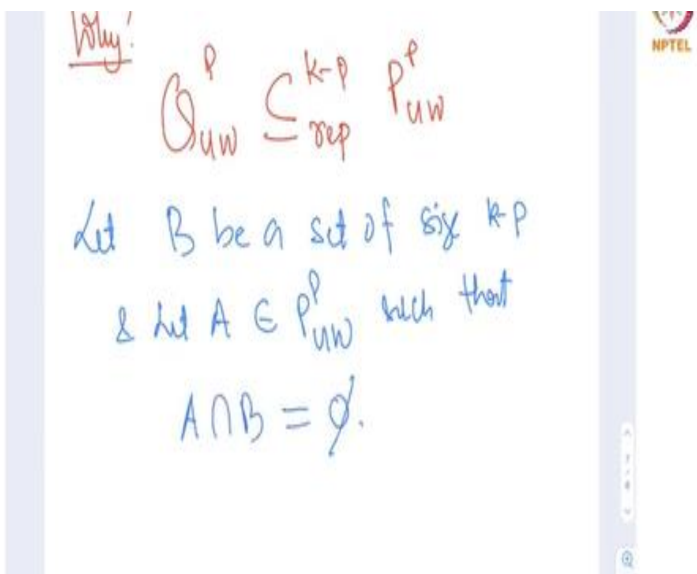
So, this is what we have done. So, now what we have, but how do we have computed is P u - 1 P - 1 representative and with w, this is what we have computed claim. First and important claim is Q u w P is a representative, but what Representative? It is a K - P representative of P u w P. So, this is a K - 1 P represented.

(Refer Slide Time: 12:30)



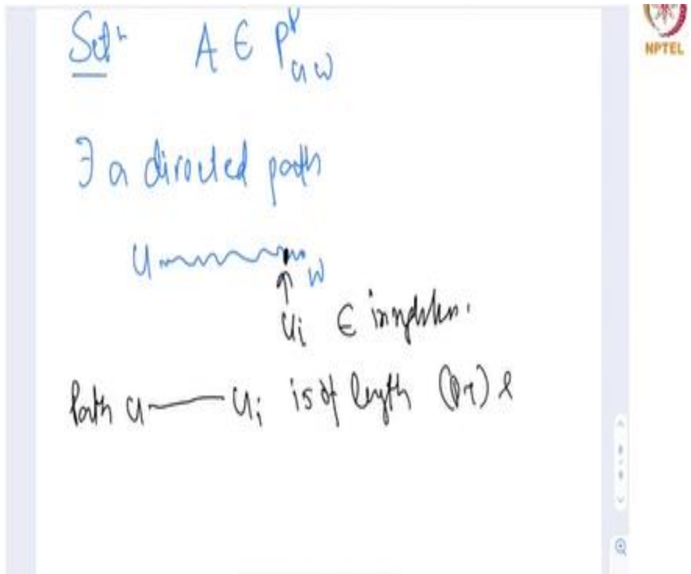
But what will be its size? So, first this and the clearly the size of Q u w P is upper bounded by, these are hat, so this is and what we compute it is? So, this is P -1 side but we computed K – P - 1 representative times in neighbourhood is upper bounded by n, so this is like P - 1 + K - P - 1 P – 1. So, this definitely less than or equal to 2 power K dot n. So, size wise this is fine.

(Refer Slide Time: 13:23)



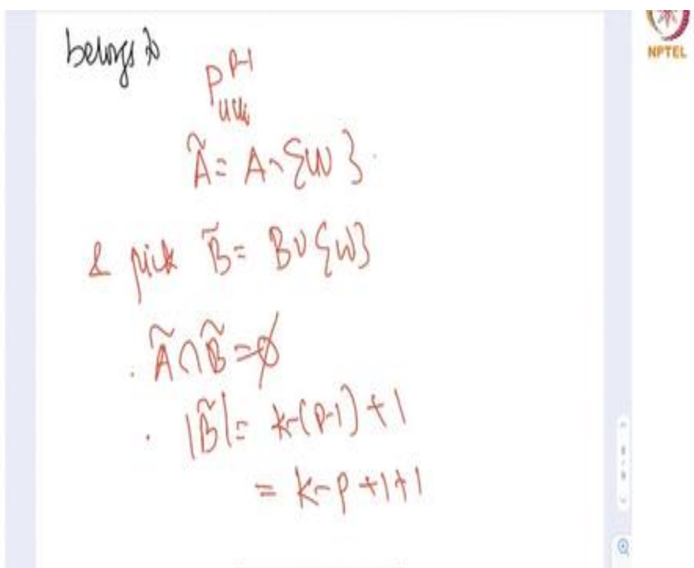
But the main question is why Q uw P is representative and that also K - P representative for P u w? Let us try to see that, why this is true? Let B be a set of size K - P and let A be element of P w, such that B intersection P P sorry, such that A intersection B is empty. What is our goal? Need to find B prime in Q P uw such that, A prime intersection B is empty.

(Refer Slide Time: 14:56)



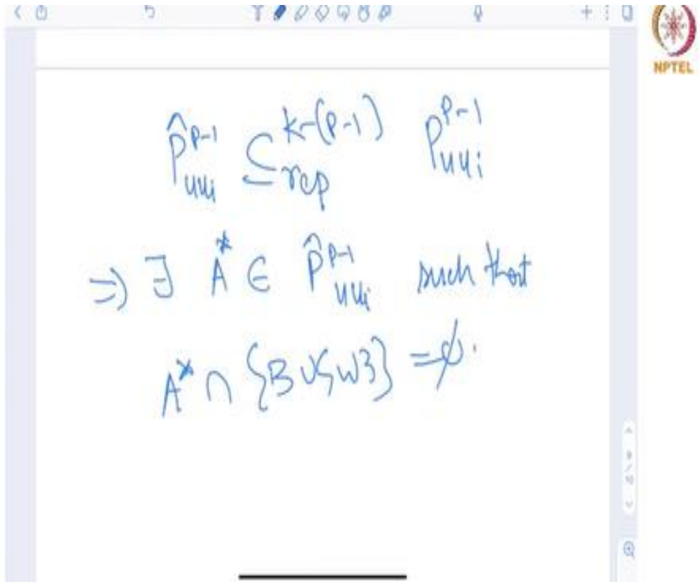
So, look at a set A, set A belongs to, so there exists a directed path u ending a W. Let us look at this. It is a direct, so this is some u i, it is in neighbour. Now, what does this implies? Implies that, the path u to u i is of length P – 1.

(Refer Slide Time: 15:55)



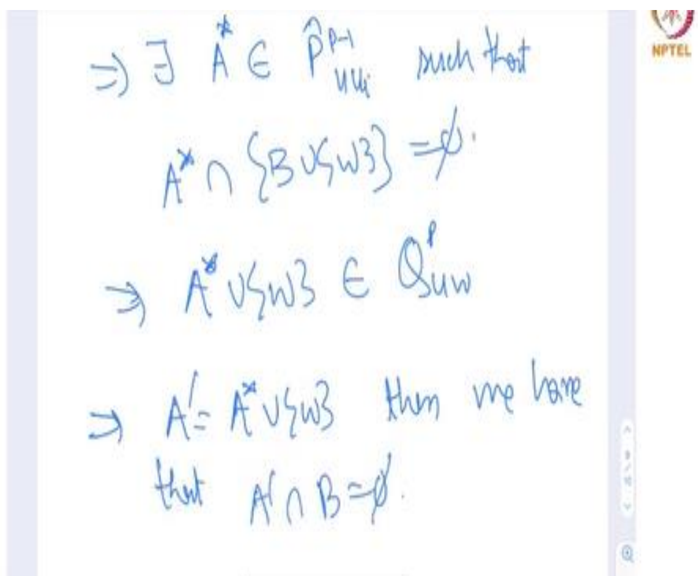
And hence belongs to be P u u i P - 1 so let us call the set A tilde is A - W. Now, and pick B tilde = B union W. Now, what is the property? The property is A tilde intersection B tilde is empty, that is the first property. And secondly what is the size of B tilde? Size of B tilde is k - P - 1 + 1 which is k - P +1 + 1, am I right? No, that is not right. This is this k - P + 1, because B tilde is size of B + 1. So, this is k - P + 1.

(Refer Slide Time: 17:15)



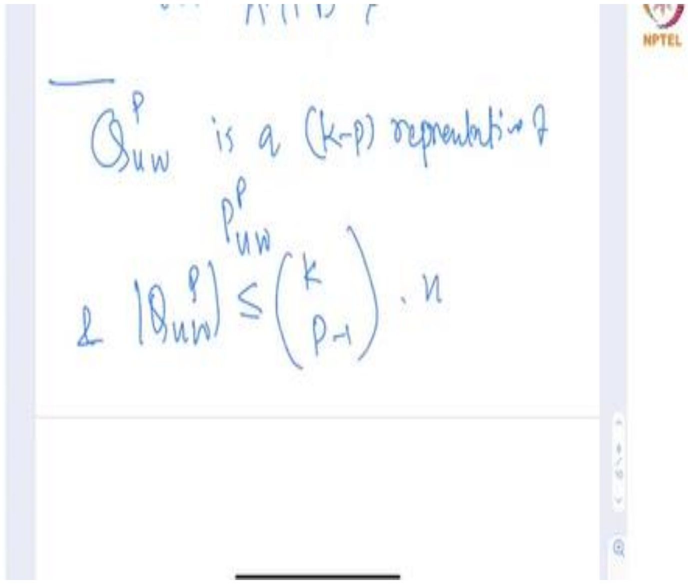
But what we know about this? We know that P hat P - 1 u u i if representative, but k - P - 1 representative of P u u i P - 1. Which implies that exist A prime that exists A star element of P hat P - 1 u u i, such that A star intersection B union W is equal to empty.

(Refer Slide Time: 18:20)



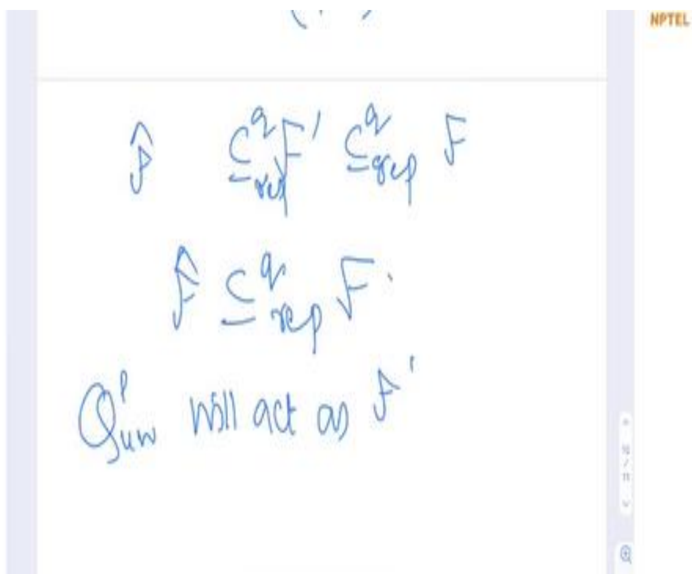
Now, but we know that A star union W belongs to Q by our construction. Because we Q P u w, belongs to Q P u w, which implies that, if I take A prime equal to A star union W, then we have that A prime intersection B is empty. So, we have been able to show that look, so when I have a singleton like you increase the size of the representativeness, that you were going to compute.

(Refer Slide Time: 19:10)



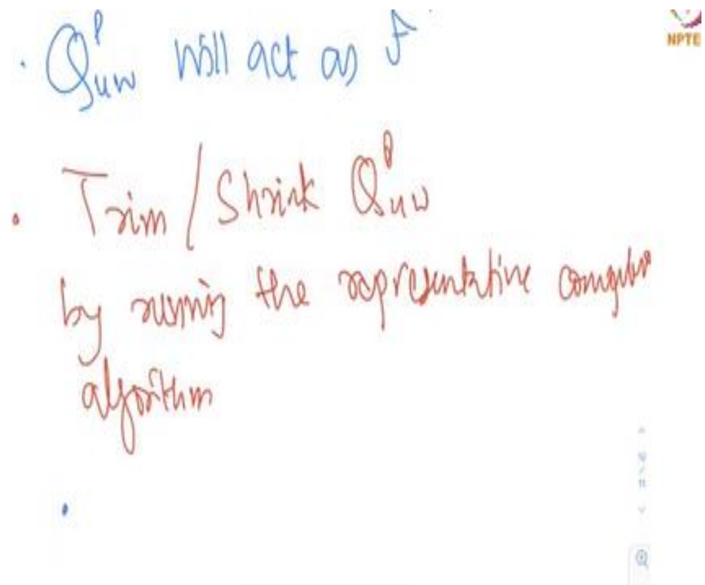
So, this shows that cube Q u w P is a k - P representative of P u w P that is great. And size of Q u w P is upper bounded by at most 2 to the power k or in fact k chooses P - 1 times n.

(Refer Slide Time: 19:49)



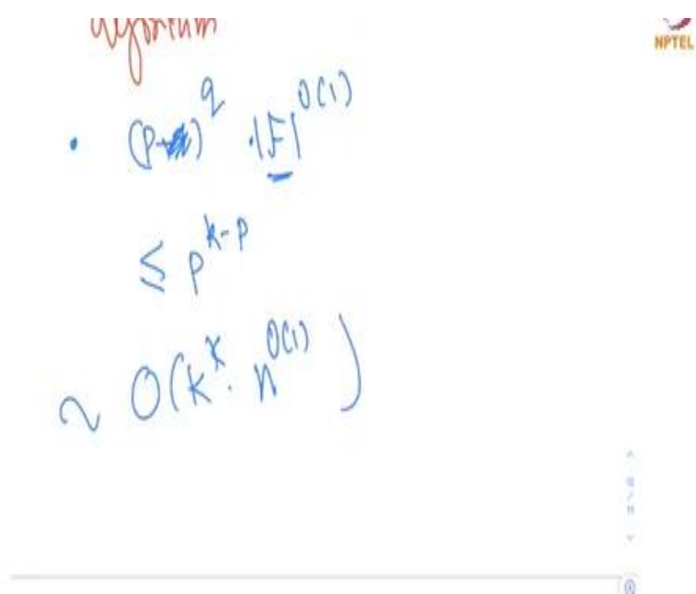
So, now what will you do? So, this is an expanse, so now this is where we will use the transitivity, you recall A, if you recall correctly that we said that look at If you are given a family F prime could you say representative Q representative of F and I have an F hat, which is also Q representative of F prime, then imply that F hat is a Q representative of F. Now Q P u w will act as F prime. And now what will we do?

(Refer Slide Time: 20:31)



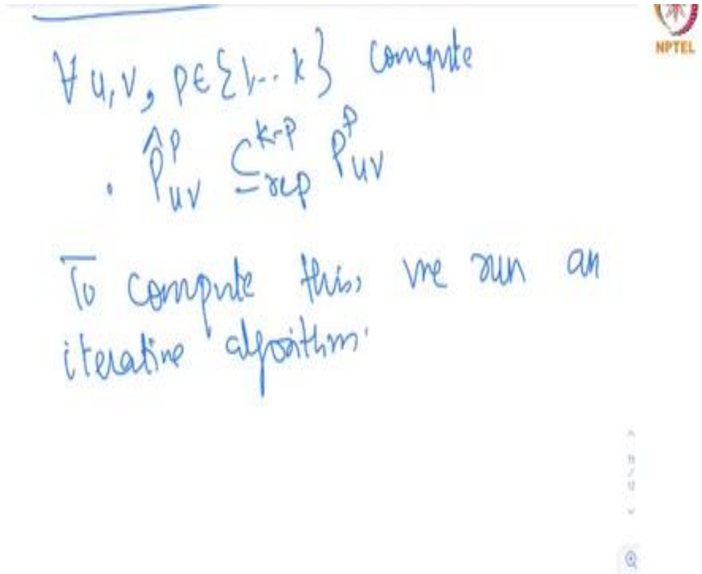
The operations we will do are trim or shrink Q u P w by running the representative computation algorithm. So, this is we are going to compute will act, so the algorithm which we had designed here, this algorithm. Now, the F which you are going to feed is a smaller Q p u w and they are going to call this algorithm. So, this algorithm is going to run.

(Refer Slide Time: 21:23)



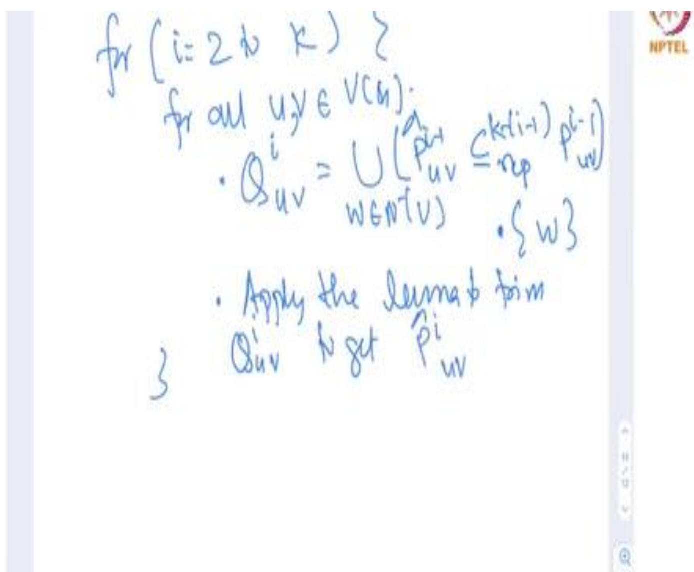
So, as such this is an algorithm which is going to run in time, if you recall it is p + q to the power q. So, in this sense it is going to run and side of the family to the power big often. So, if I forget this term, then this is the algorithm which is going to run in time, in fact this is p to the power q. So, this is going to be a run in time p to the power k - p roughly. And, so basically this will give you an algorithm with running time k the power k n to the power of 1. So, what is an algorithm?

(Refer Slide Time: 22:04)



Algorithm is very simple, the basic principle for all u v and p in 1 to k, compute hat. For each compute what k - p representative of the set P u v p, and how do we compute this? To compute this, we run an iterated algorithm. To compute this, we run an iterated algorithm, what is iterated algorithm we do?

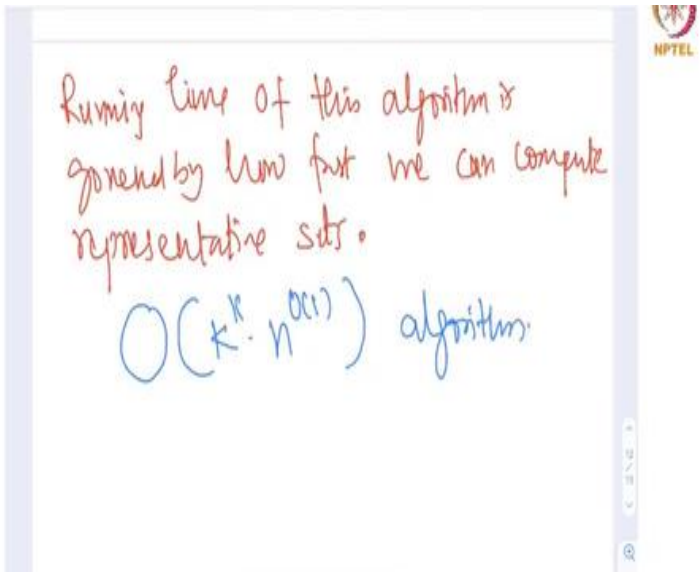
(Refer Slide Time: 23:10)



So, first we compute like F, so for i = 2 to k, what did we do? We had the following algorithm. First for i = 2 to k, we are going to compute whatever that, we are going to compute P i u w for all u v in the G. What are we going to compute? We compute after you in a slice, you compute Q u v i by going over all w in a neighbourhood of v and computing, for which you have computed P i - 1 representative u v, you have computed which is a subset of and what is the representative?

It is k - i - 1 representative of P i - 1 u v dot w. So, this is what you have computed and you do this, and then apply the lemma to trim Q i u v to get P i u v hat. So, this is what the algorithm does. So, you notice given so, what is this step?

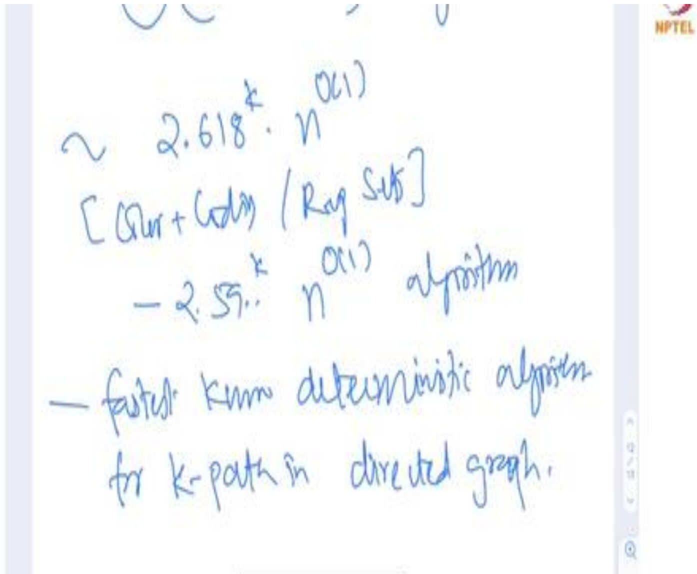
(Refer Slide Time: 25:29)



So, this is a expand step. What is this? Shrink step. So, you iterative leave for each you view, what you do? Expand they knew it is followed by a shrink operation. It is expand shrink expand shrink, expand shrink, expand shrink and this is how you can compute the all the desired set and if for any u v, P hat k u v is non-empty, you have got a path. So, this is an algorithm and the running time of this algorithm is governed by, how fast we can compute the rep sets, how fast we can compute representative sets.

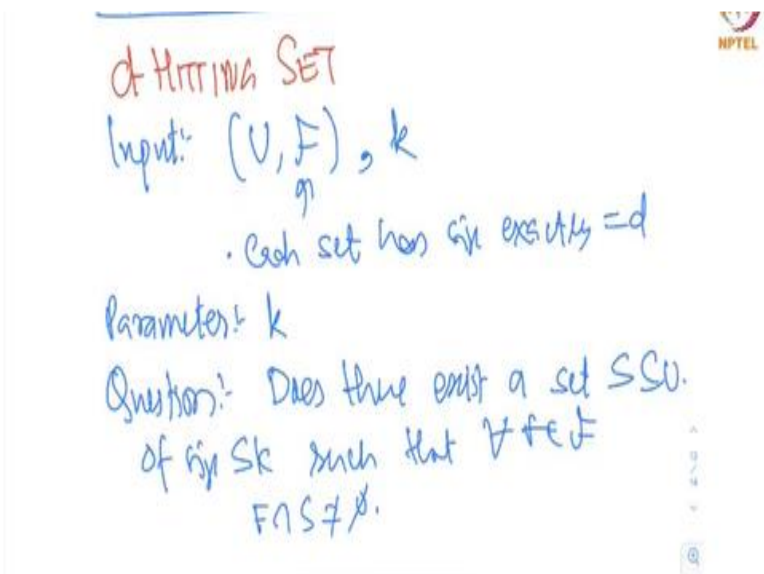
There are much-much faster algorithm to compute representative set. But if you apply this naïve algorithm, we saw that we can get a k power k times n to the power big O of 1 algorithm.

(Refer Slide Time: 26:58)



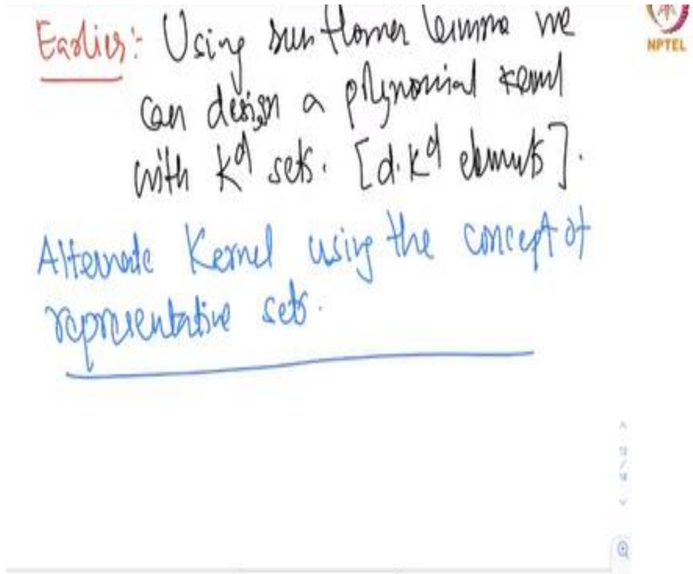
But as I said, there are very, very fast algorithm for this and using that fast algorithm, you can actually compute this in time 2.618 to the power k to the power n to the power the big 1 and applying more sophisticated technique like colour coding combined with Representative sets. We can get slightly I think 2.59 dot, dot sum to the power k into the power big O for algorithm. And this method gives the fastest known deterministic algorithm for k path in directed graph. So, this is an application one we saw.

(Refer Slide Time: 27:58)



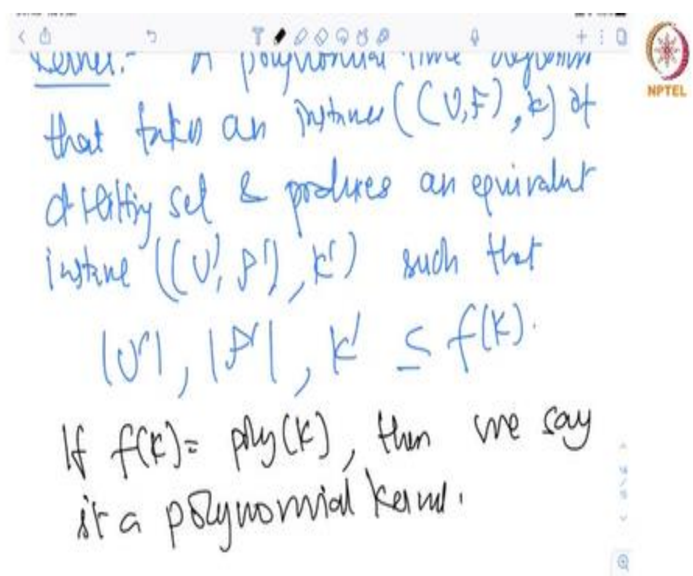
And let me give you another simple application is the following problem, d hitting set. Input is a universe, a family, an integer k, what is the property of family? Say each set at size exactly equal to d. This can also handle at mostly but like just for illustration purposes is k. And our question is, does there exist a set S subset of U of size at most k, such that for all F in curly F, F intersection S is not empty. So, it is a d hitting set.

(Refer Slide Time: 29:20)



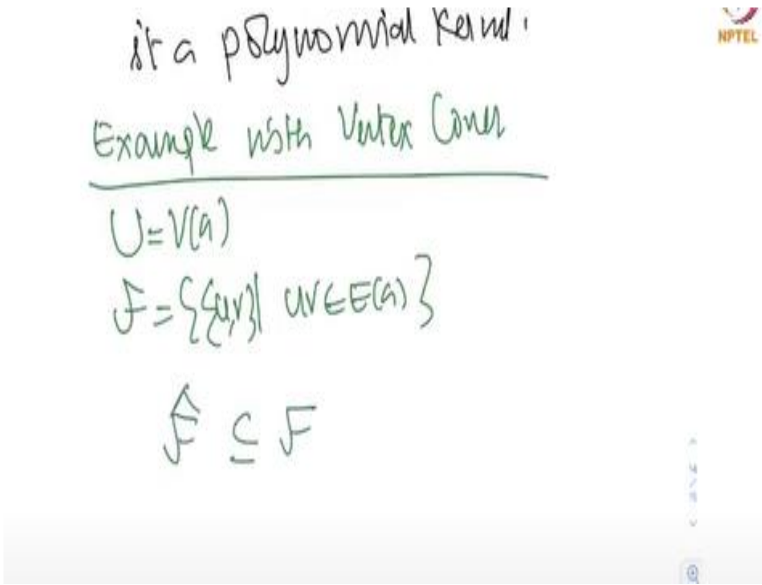
So, what we have seen before earlier we have seen, earlier what we have seen? Using sun flower lemma, we can design a polynomial kernel of size k to the power big O of d. In fact, of size of kernel with k power d sets and that automatically implies d into k power d elements. So, we will now what we will try to do? We will give an alternate kernel using the concept of representative sets, using this.

(Refer Slide Time: 30:37)



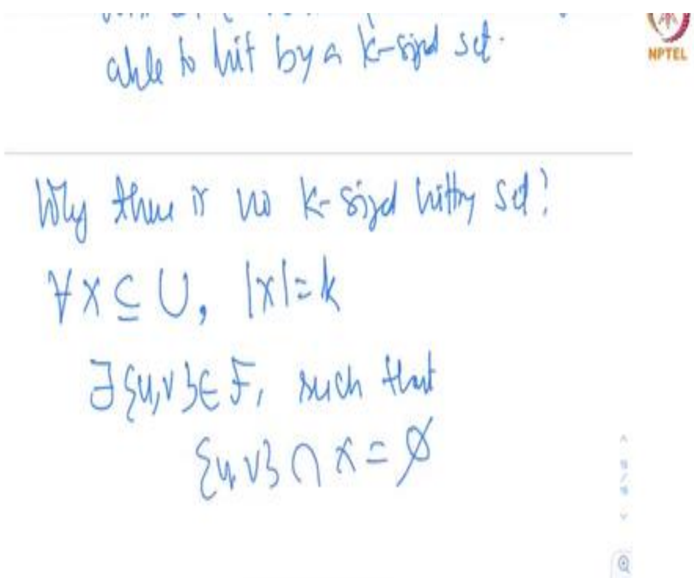
So, first of all maybe if you have forgotten, so let me repeat what kernel means? It is a polynomial time algorithm, so it is a polynomial time algorithm. That takes and instance U, F, k of d hitting set and produces an equivalent instance u prime, f prime, k prime, such that u prime f prime, k prime is upper bounded by some function f of k. If f of k is some polynomial in k function, then we say it a polynomial kernel.

(Refer Slide Time: 32:00)



And I think for just for illustration purpose, first let us do example with vertex cover. So, how I am going to treat my vertex cover? So, now my U is V G and my family is u v, u v is in my edge. Now, I want to compute F hat subset of F.

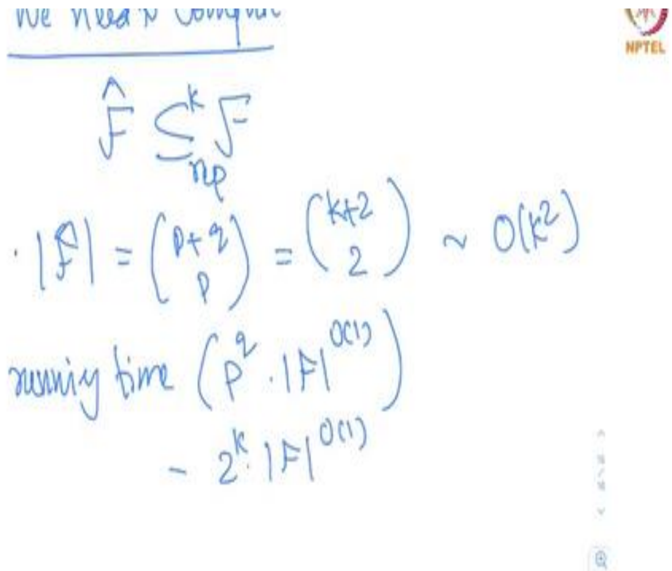
(Refer Slide Time: 32:43)



So, let us ask ourselves what we are trying to achieve from F? b So, what would you like to achieve from F? So, F hat will store reason for F not being able to hit by a k sized set. So, or rather, let us ask ourselves why there is no k sized hitting set? Which implies that for all x subset of U, mod x = k, there exist u, v in F, such that u, v intersection with X is empty set. So, now the moment we turn this into this kind of question.

You notice that this has a flavour of representative set. So, now what I am going to come to, what I am going to call that? So, what this means that my family needs to have a disjointed property with respect to any case I said, this is what it needs to pay your, which implies that what should I compute?

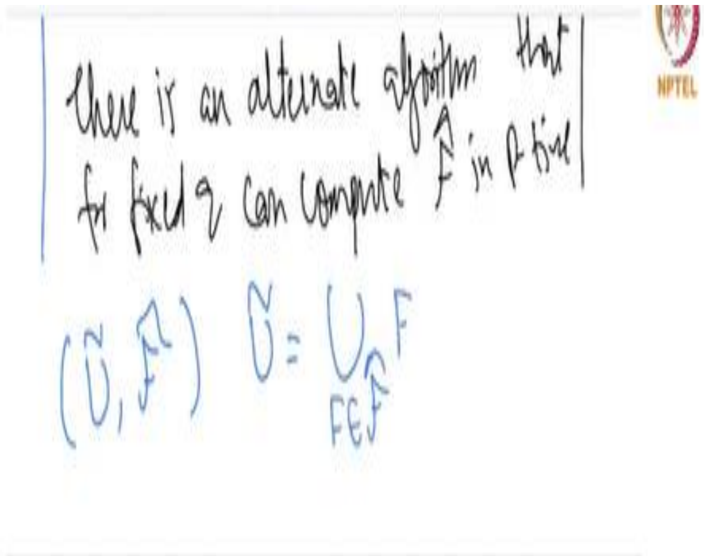
(Refer Slide Time: 34:15)



We need to compute the following, you will compute F hat, which is subset of F, it is a representative, but it is k + 2, representative, this is what I am going to compare. So first of all, what is the side of F hat? Well, the size of F hat if you recall, if p + q choose P, which is k + 2, p + q is k + 2 and P itself is 2, so this is order k square. And what is the running time of this algorithm?

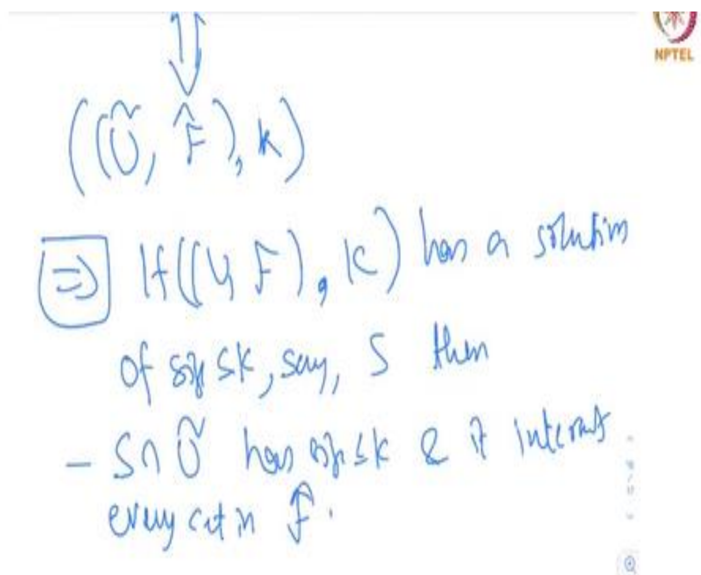
Running time of this algorithm, if you recall correctly, it is P to the power q are the making correct. So, let us go back, so we have to compute k representative because we want to be. So, it is this, but if we apply our running time naive algorithm, then this is if you notice this is going to be 2 to the power k.

(Refer Slide Time: 36:17)



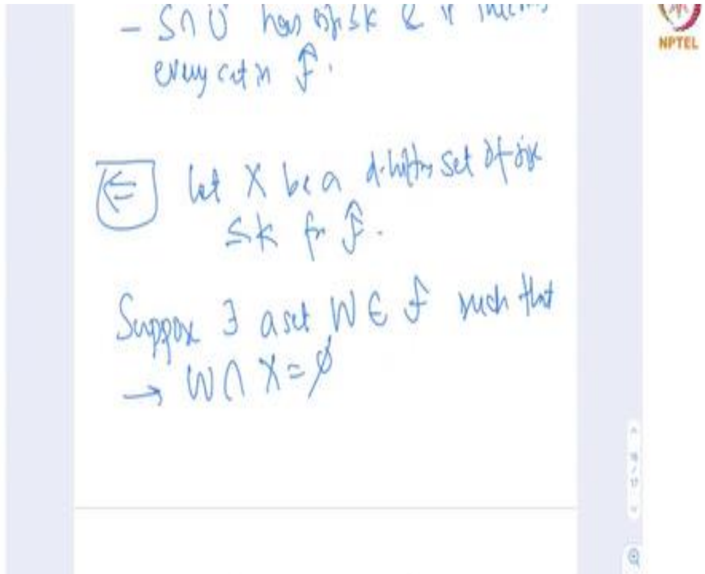
So, there is an alternate algorithm that for fixed q can compute F hat in Polynomial time. So, we have computed this. Now what is our kernel? I will say look at U tilde F hat, and what is U tilde is basically endpoints of or rather let us say, so all the union of the sets, now that is it.

(Refer Slide Time: 37:19)



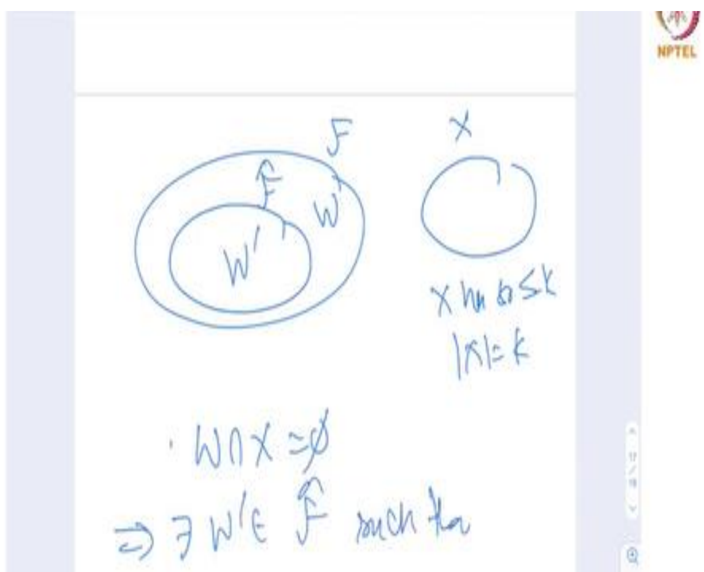
Now notice that we have U F, k, I want to show this is same as. So, forward direction is fine, why? Because, if U, F, K has a solution upside at most k say S, then S intersection U hat had size at most k and it intersects every set in F hat.

(Refer Slide Time: 38:19)



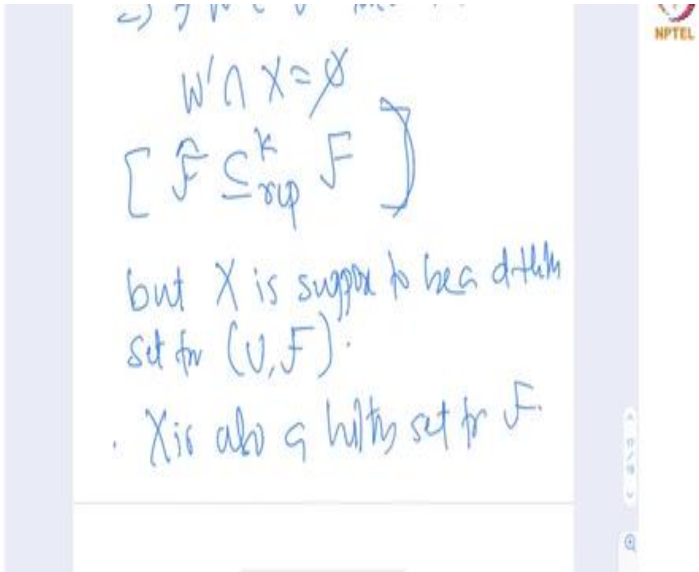
Let us look at the reverse direction. Let X be a d hitting set of size at most k for U tilde F hat, k rather for F hat. Suppose, there exists a set say W in F, such that W intersection, F W intersection X is empty, meaning look if X would have been a d hitting set for F like U, F also then we are happy. But suppose W is not a d hitting set, it means you can find a set W, so the W intersection X is phi.

(Refer Slide Time: 39:25)



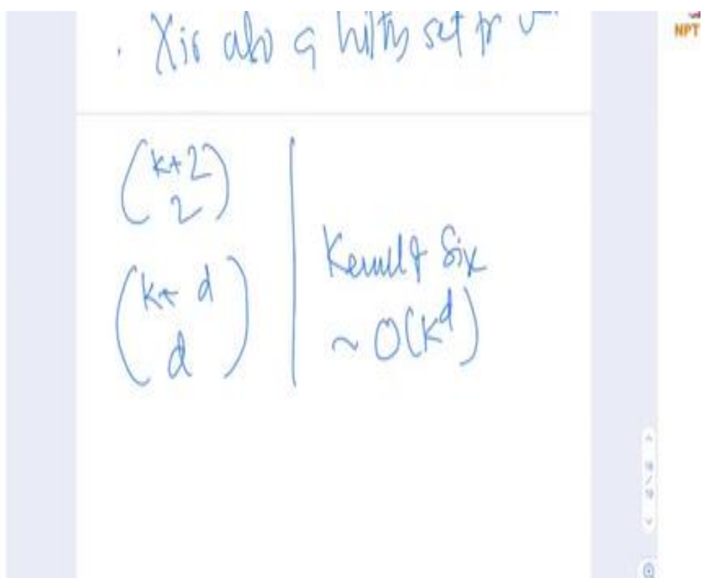
So, now let us look at this, here your family, here is your F hat, and you have a set X, X has size at most came, I mean you can assume that X has size exactly equal, it does not matter, because by adding some this. Now, you have found a set W in F. W intersection X is empty, implies there exists a W prime in F hat such that W prime intersection X is empty.

(Refer Slide Time: 40:09)



And why this is true? Because F hat is a representative with respect to any case I said of F. So, this implies that W prime intersection X, this is but X is supposed to be d hitting set for F. This implies X is also a hitting set for F that is it.

(Refer Slide Time: 40:49)



So, now we have got a kernel, so for vertex cover we got a kernel of size k + 2 choose 2 and but for d hitting set you will get k + d choose d. Because, the family has a d sized set and you want disjoint as property with respect to any case I said. So, this also gives you a kernel of size roughly kth power d. So, this is a second application for, second and another simple application of representative sets.

(Refer Slide Time: 41:27)

So, we will see another application but for directed at least K cycle in the last lecture and also some more and it is extensions to what is Matroids and that is where the name comes and where all these things fit in, so that we will do in the next and the final class.