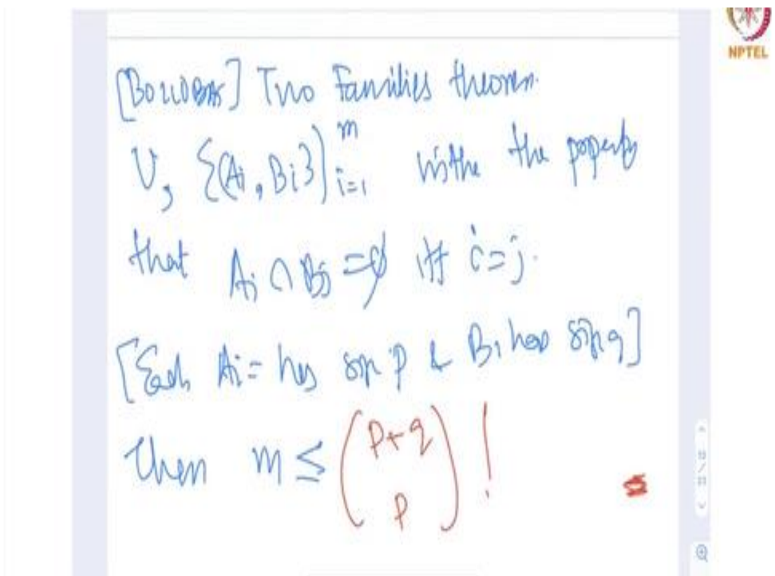
Parameterized Algorithms
Prof. Neeldhara Misra
Prof. Saket Saurabh
The Institute of Mathematical Science
Indian Institute of Technology, Gandhinagar

Lecture – 44
Matroids: Representative Sets – Computation and Combinatorics

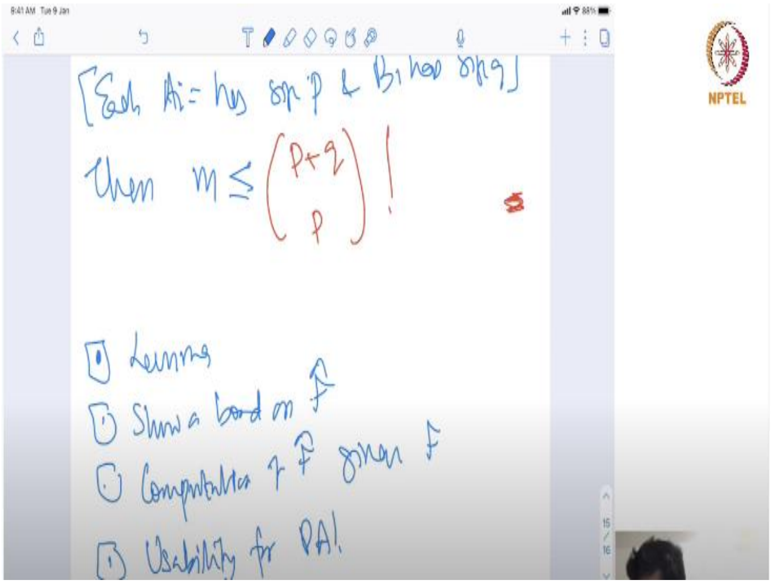
Welcome to the second lecture on matroids. So, last time we set up a kind of definition intuitively taken from the directed k path algorithm.

(Refer Slide Time: 00:27)



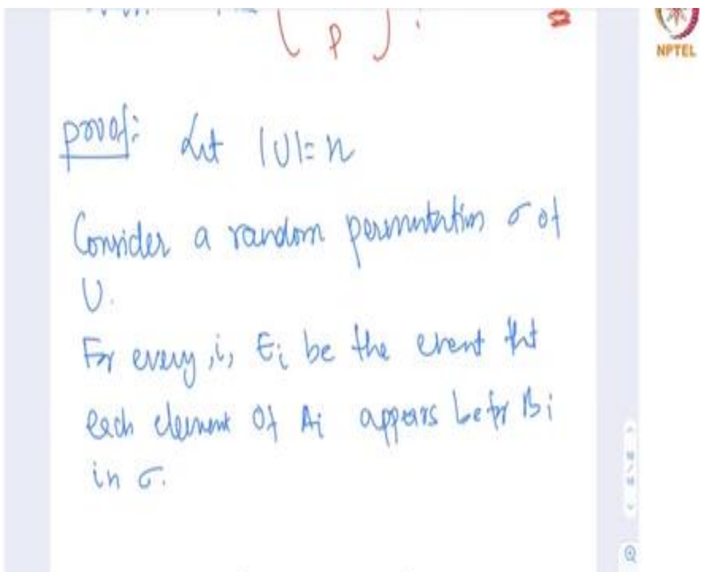
But before we go, this is a two Bollobas family of lemmas we were talking about last time. So, given a universe and the pair of sets A i, B i with the property that A i intersection B i is equal to empty if and only if i = j then the number of such pairs that we can obtain is p + q choose p.

(Refer Slide Time: 00:47)



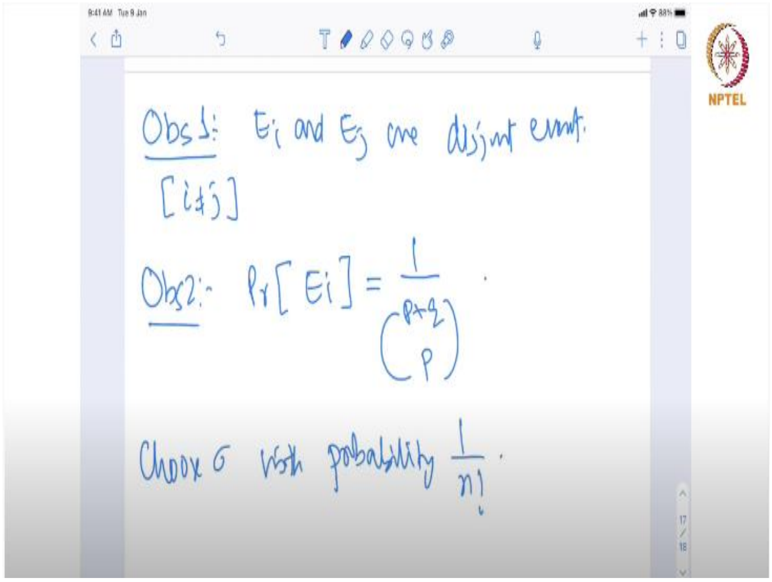
And the plan was that we will give a proof of this lemma and see how we can derive what you call the cardinality F hat which is same, how to compute this and then its usability.

(Refer Slide Time: 01:02)



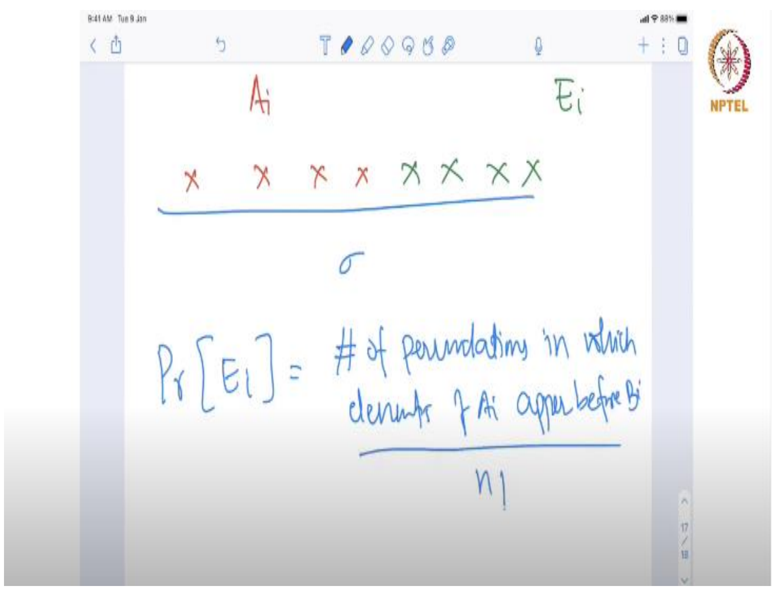
So, now let us just boil down to proof of this lemma. So, let us prove this lemma first. So, here is a proof. Let us consider a random permutation. Consider a random permutation sigma of U. And for every i in m, two lemmas we would like to prove. So, this defined for every i E i be the event that each element of A i appears before B i. So, this is my going to be my event. I will explain to you. So, we are going to prove two lemma or two observations.

(Refer Slide Time: 02:28)



Observation 1 is that E i and E j are disjoint events. Of course, i is not equal to j and observation 2 is that probability of E i that event E i happened is equal to 1 over p + q choose p. So, now let us try to see what is the random experiment happened. So, we chose a random permutation rho of U meaning there are n factorial permutations of U. You chose one permutation rho. So, you chose a rho with probability 1 over n factorial.

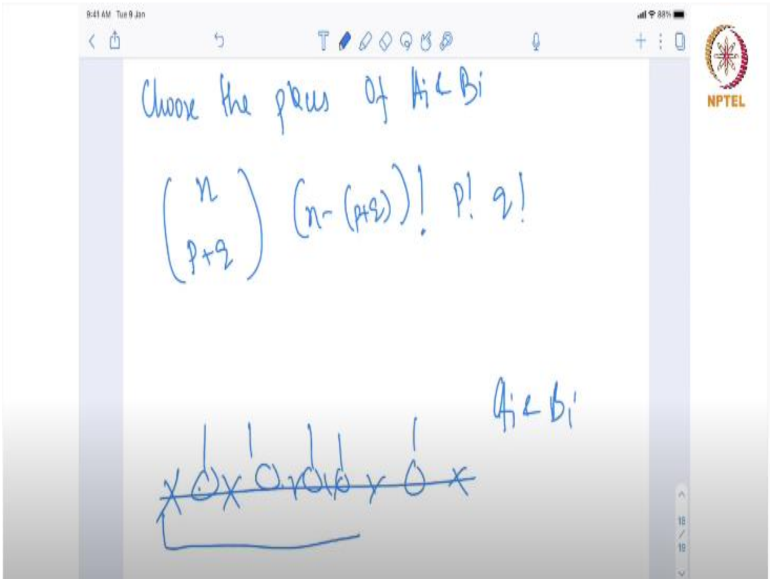
(Refer Slide Time: 03:33)



And what is my event? So, event is look at this permutation rho. Now it means you mark the elements of A i. They are the elements of A i and we say that look and suppose the red crosses are element of say A i and let us suppose the green crosses are going to be elements of B i. So, I am going to call that what is an event if all the red crosses meaning you mark all the vertices which are from A i as red and all the red crosses appear before green crosses.

Then that is your event E i. If there is any red cross after some green cross then that is not a correct event. So, now you ask yourself what is the probability of happening of roh? So, what you do first? So, now let us try to understand how many permutations are there. So, what is the probability of the event A i? So, probability of E i is number of permutations in which elements of A i appear before B i divided by n factorial. So, how do we do this?

(Refer Slide Time: 05:24)

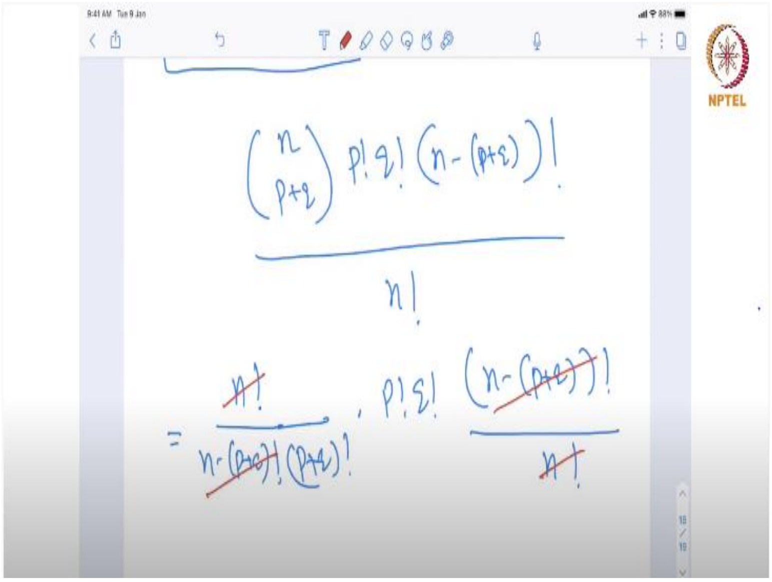


So, first of all you choose the places of A i and B i. How many such places are there? See from this you choose n choose p + q. So, you have chosen. So, first you have chosen and you know that on the remaining like on these places p plus the elements of so you chose some. So, you know that these are allocated for elements of A i and B i and on the remaining any element of my universe can come. So, that is n - p + q factorial.

And now look at the places where I know that the out of these p + q places first p places are given for A i and the next q places are given for B i. But how these elements are permuted among themselves it does not matter to us. So, this is p factorial in q factorial. So, this is how you can have a number. So, these are the number of permutation which elements of A i appear before B i. You choose the places where A i and B i will occur that is p + q.

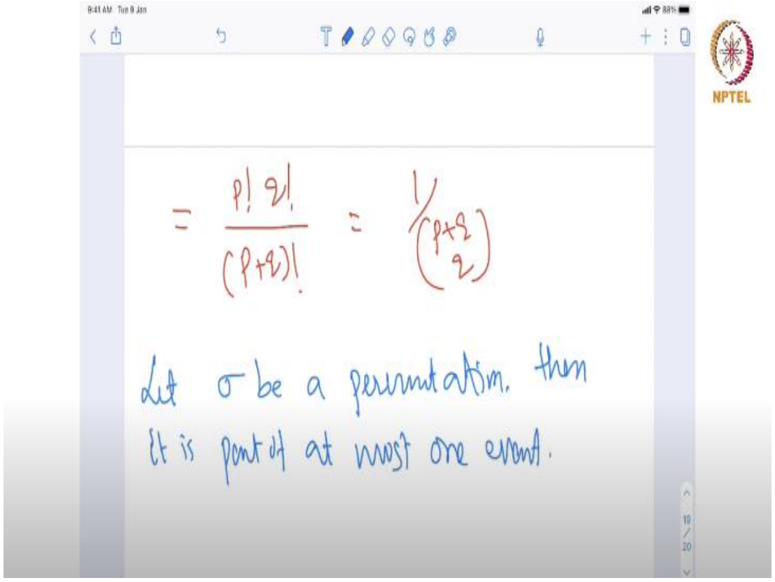
You know the first places are taken by A i and the last place they are taken by B i. And how they are permuted among themselves does not matter. So, p factorial into q factorial and in the remaining places any of the elements can occur. Any permutation of the remaining element can occur which is n - p + q factorial.

(Refer Slide Time: 06:59)



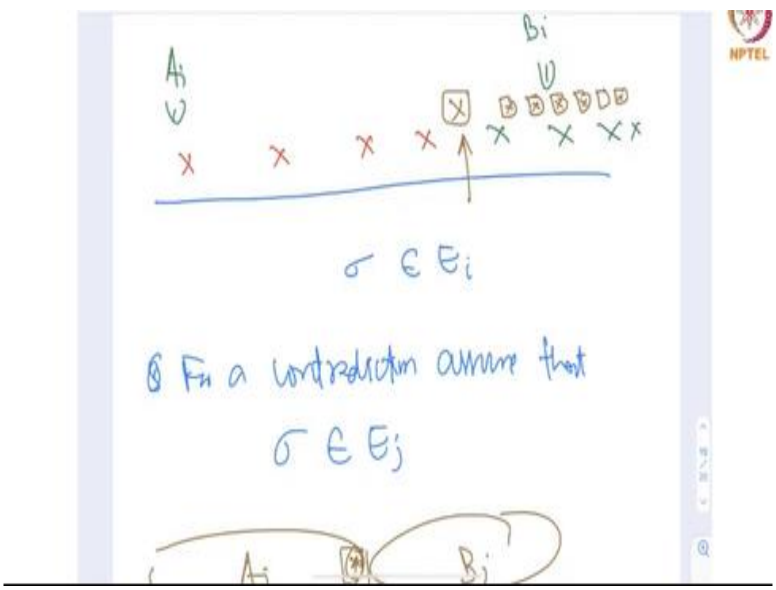
So, now let us do this computation. So, what are we going to get? n choose p + q p factorial q factorial n - p + q factorial divided by n factorial. So, if you do some amount of computation which I am not very great at but I could try, n factorial from here this is n - p + q factorial and p + q factorial times p factorial times q factorial times n - p + q factorial divided by n factorial. So, this and this term will get cancelled, this term and this term will get cancelled.

(Refer Slide Time: 07:57)



And what you are left with is precisely p factorial q factorial by p + q factorial which is nothing but 1 by p + q choose q. That is, it. Check, so that is it. So, we first proved observation two. Now observation say that E i and E j are disjoint event. What does mean by this? What is the meaning of this? Let sigma be a permutation then it is part of only this is part of at most one event. Then this is part of at most one event.

(Refer Slide Time: 09:02)



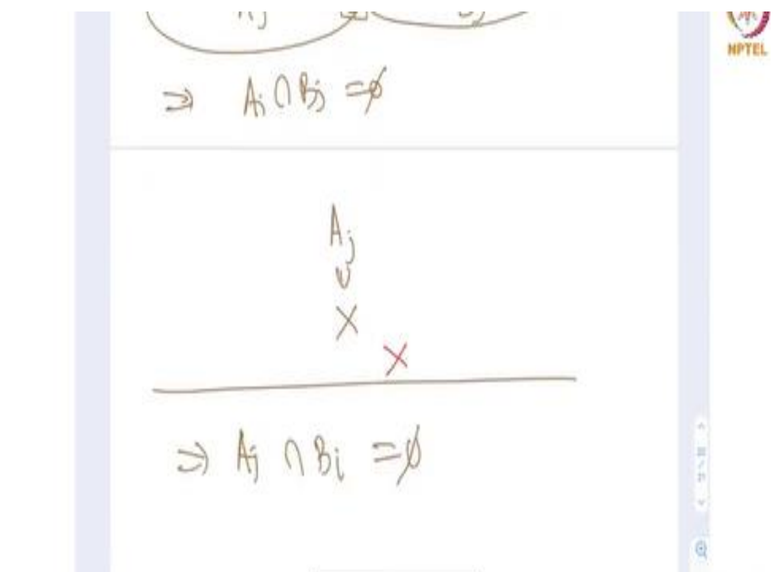
Now let us look at this. So, what you know? So, let us suppose you fix a rho and suppose we had this is event E i. So, what is the meaning of this that all the rate crosses of E i occurs here and all the green crosses are element of A i and these are element of B i. Now let us look at any other contradiction assume that sigma is also part of some E j. Now let us try to understand where will my let us draw them by brown. So, this is what do you know about E j.

You also know that if sigma is in E j then all the elements of A j occurs all the elements of B j. Now let us look at the last element here. Where can it be? So, let us call this element brown, can this element occur here? Because if this element occur after this last red cross, guy, then what happens? All the other elements of B j are going to be here. All the elements of this guy, is going to be here.

If the last element of A j either occurs after the last red or it does not, we will see. But if it occurs at the last element of this then what happens is that look at all the elements of brown. They are the all the elements of B j. They also occur after this which implies that the red crosses and brown crosses like this and these elements if this will happen then that will imply that A i intersection B j is equal to empty.

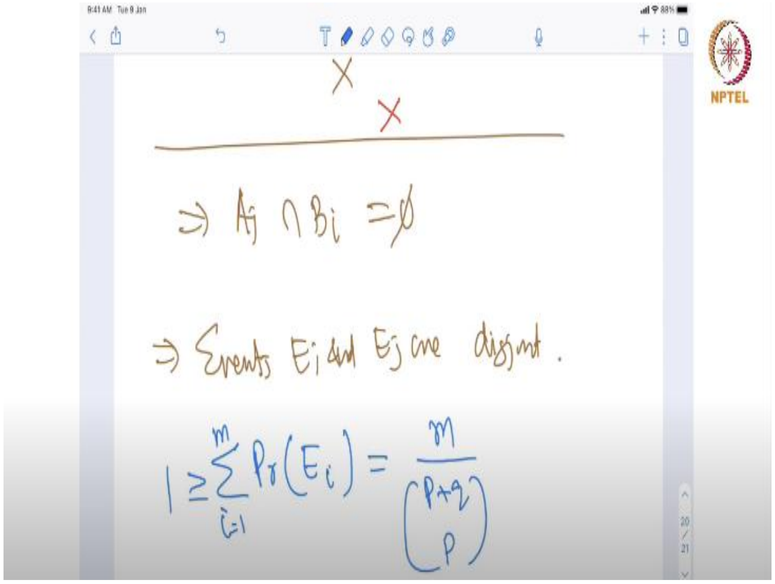
But that cannot happen. So, you know that last element of A j cannot occur after this. So, it should occur somewhere before.

(Refer Slide Time: 12:00)



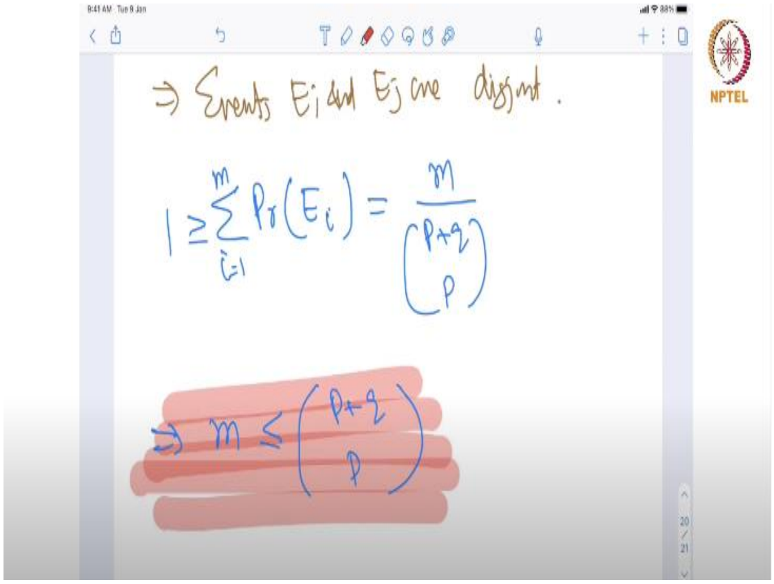
So, imagine this is my last red cross and suppose my brown cross occurs here. Let us see somewhere here. So, this is an element of A j. But if this happens then what is an intersection of if this happens that will imply that A j intersection B i is empty. So, if the brown cross occurs after the last red cross, then we can say that A i intersection B j is empty. But if A j occurs before the class, then you can say A j intersection B i is empty.

(Refer Slide Time: 12:56)



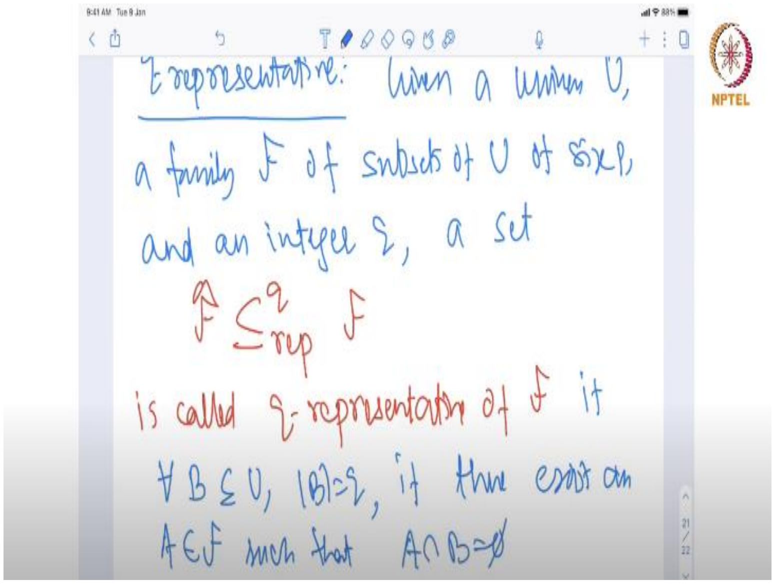
But in both cases, we get a contradiction which implies that events E i and E j are disjoint. Now why are we interested in it? So, look at this probability of E i summation i going from 1 to m is at most 1 and this is equal to m times p + q choose p. So, what this implies?

(Refer Slide Time: 13:45)



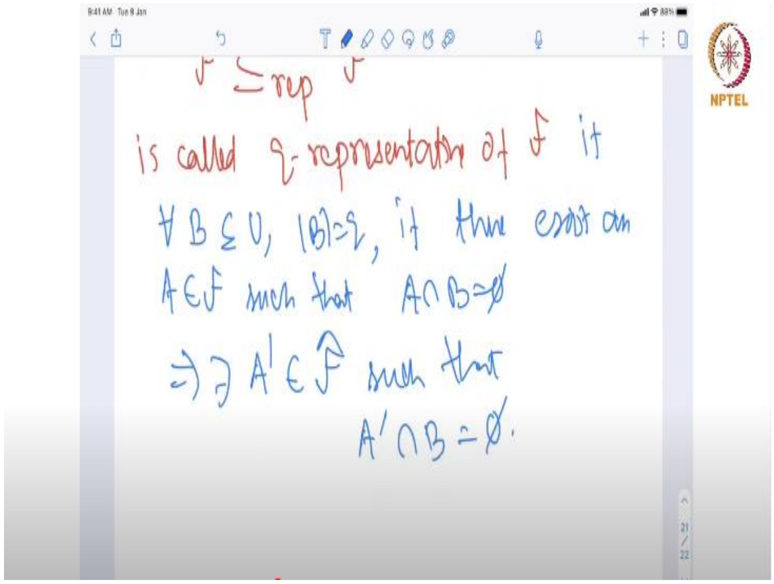
This implies that number of such sets are upper bounded by p + q choose p and that is it. So, the first thing we have been able to show is that we have been able to prove this lemma which tells us that below that the number of such sets are this.

(Refer Slide Time: 14:20)



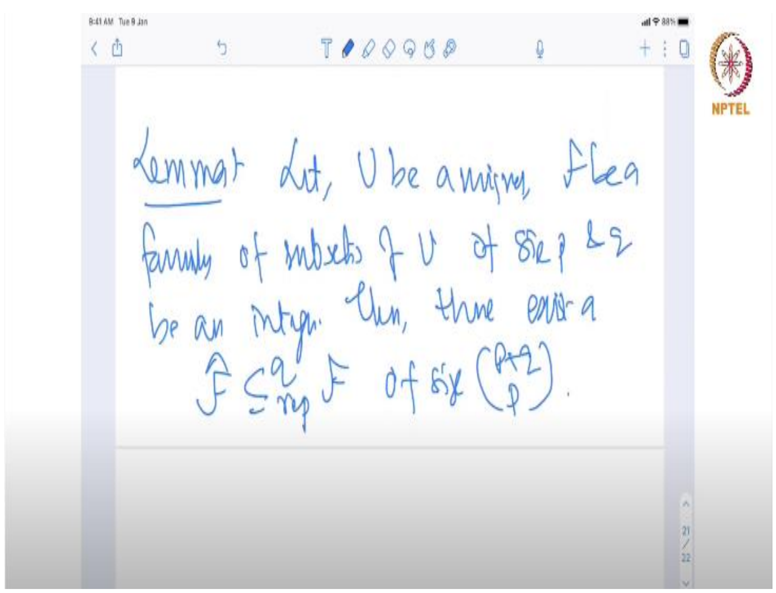
Now given this let us try to phrase everything properly and define this notion of q representative family. What is the q representative? Given a universe U, a family F of subsets of U of size p and an integer q a set F hat is called q representative of F, if what is the property? F hat is a q representative of it. So, we already define the notion of q representative is basically if for all B subset of U or rather if for all B subset of U if there exists an A in f such that A intersection B is in phi.

(Refer Slide Time: 16:19)



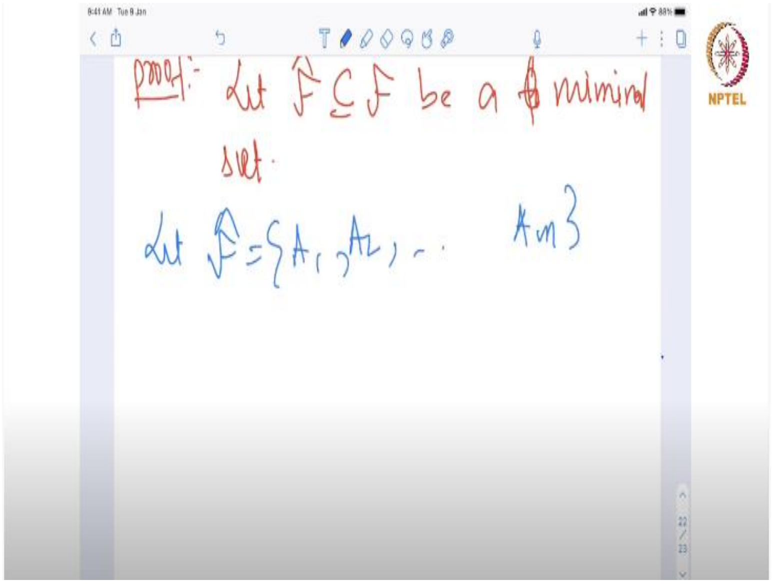
This implies there exists A prime and F hat such that A prime intersection B is phi. So, I am going to call a q representative if it is respects disjointness with respect to all queues I said. In the sense that if I give you a B and you have some set A in F such that B and A are disjoint and B has size q then you have to guarantee that such a set exists in F hat.

(Refer Slide Time: 16:54)



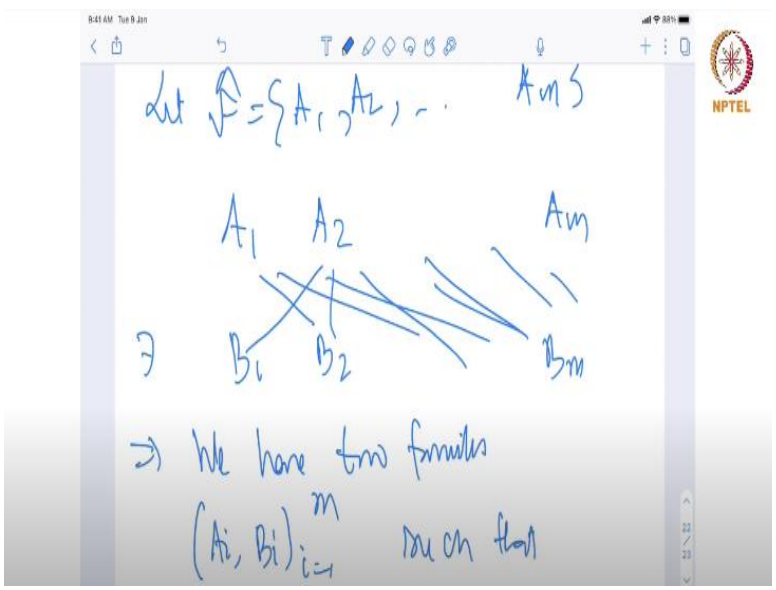
And what did we just prove? So, this is like another lemma that we just now showed. Let U be a universe, F be a family of subsets of U of size p and q be an integer. Then there exist F hat q representative of F of size p + q which is p and the proof is now simple.

(Refer Slide Time: 17:47)



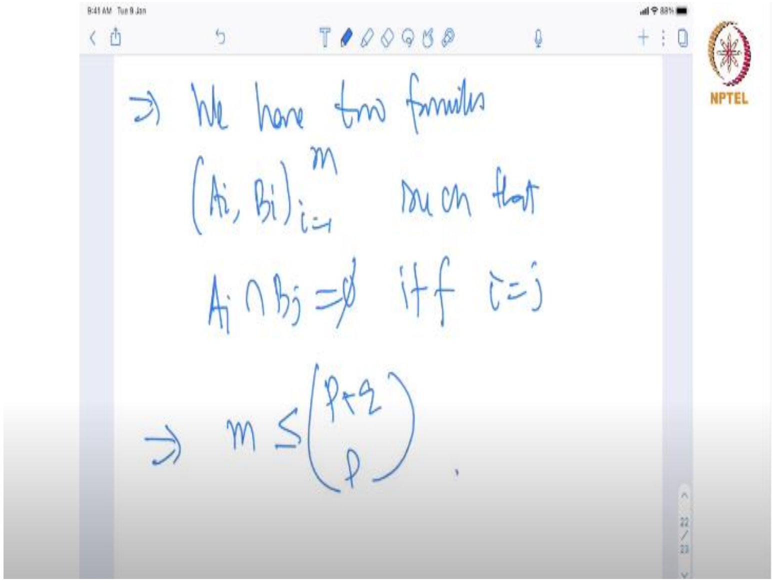
What is the proof? We already have defined the notion of minimality. Let F hat subset of F be a minimal set. Now if it is a minimal subset what you know and let F hat be A 1, A 2, A m.

(Refer Slide Time: 18:25)



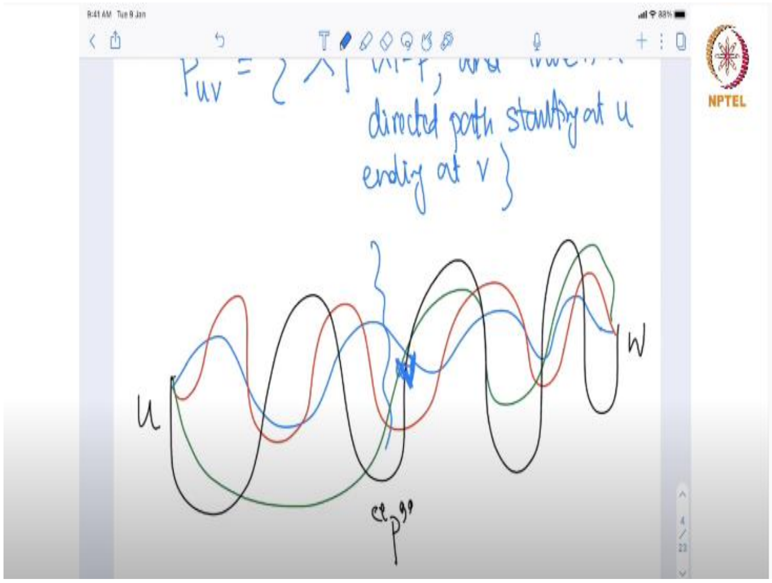
Now since F hat is a minimal you will be able to find for each A 1 B 1 which intersects with every other A 2. So, basically why they are minimal? Because there exists B i with this property such that whatever property that A 1 intersects with every other B m A 2 intersects with every B 2. This implies that we have two families A i, B i, i going from 1 to m.

(Refer Slide Time: 19:19)



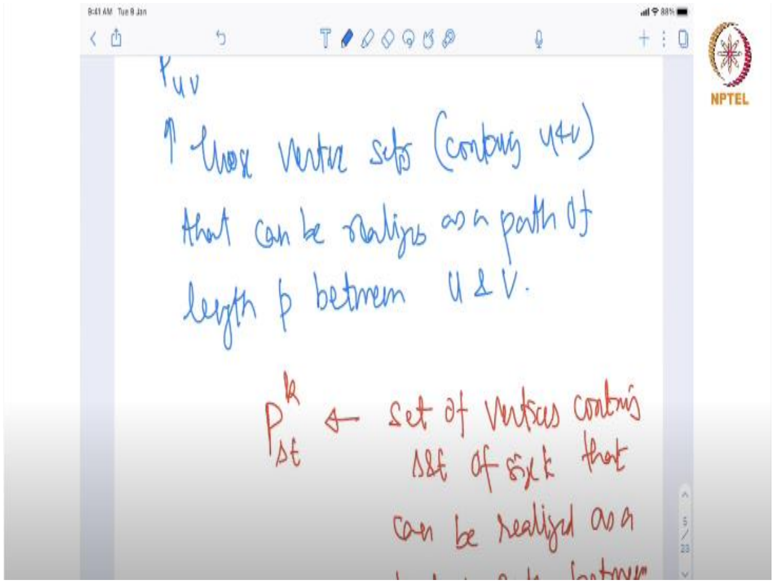
Such that A i intersection B j equal to empty if and only if i = j. Now what does that imply by two families theorem? This implies m is p + q choose p. So, now we have been able to show. So, let us go back and let us trace.

(Refer Slide Time: 19:42)



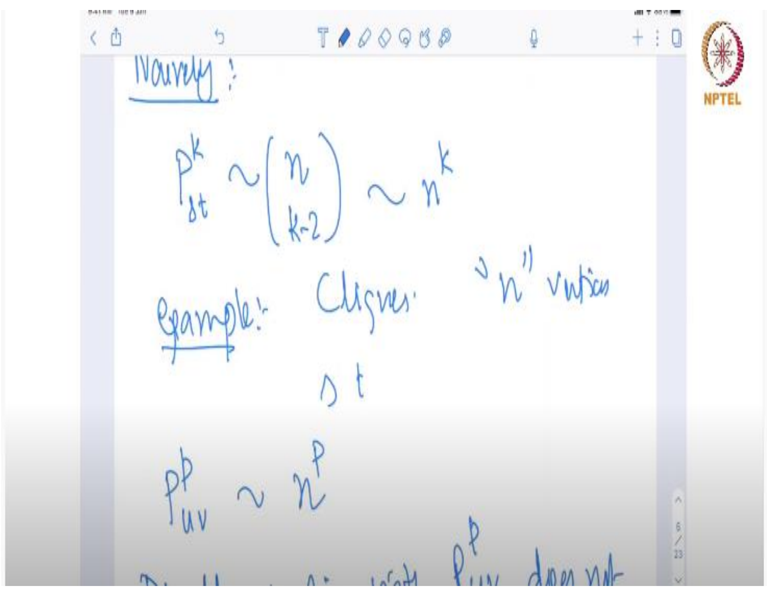
So, we introduce this notion of like if you are trying to understand or trying to construct a directed path algorithm, for example.

(Refer Slide Time: 19:51)



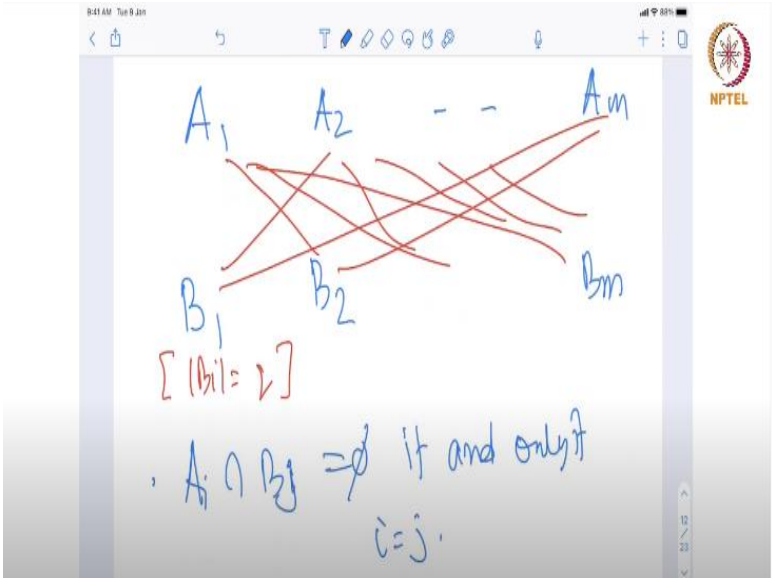
Then if you look at the family of partial sets which could be like these are like those sets which contain a path from u to v of certain length. And basically, you can think of these like prefixes of the path of length k that we are trying to construct.

(Refer Slide Time: 20:06)



And we said look that is too many but look why do we need to keep all this? Let us try to keep only those sets that are important. This set is important because there is some set which it is disjoined with and no other set that I have kept in the family can be disjoined with this. Otherwise, if for every set that this guy disjoined with someone else is already disjoined within my family then this set is useless for me, I could have thrown it out. So, that is because of this that we got to the notion of what I call representative family.

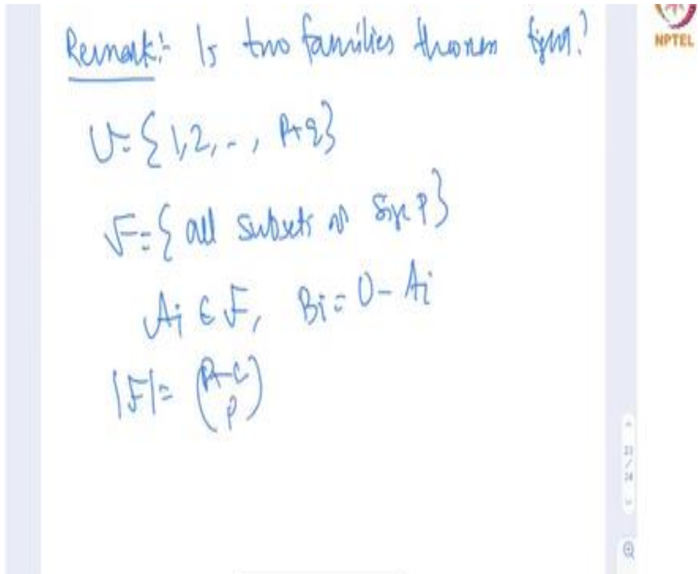
(Refer Slide Time: 20:45)



And the heart of the q represented family was this two families theorem. That basically are given a universe A i B i and the properties then the number of such sets that we could have is upper bounded by p + q choose p. And then we just now saw the proof for this that this is true and then we connected it then we define the notion of represent. This is the most important definition so what is that you are given a universe.

You are given a family of some set size B and you would like to compute a sum set F hat subset of this. What is the property? That with respect to every set of size q if F had something like it satisfies what I call disjointness property. Meaning if you give me a sum set B of size k so that you can find me a set A in my family so that A intersection B is phi. Then I should be able to give you A prime in the set which I have kept so that A prime intersection B is phi.

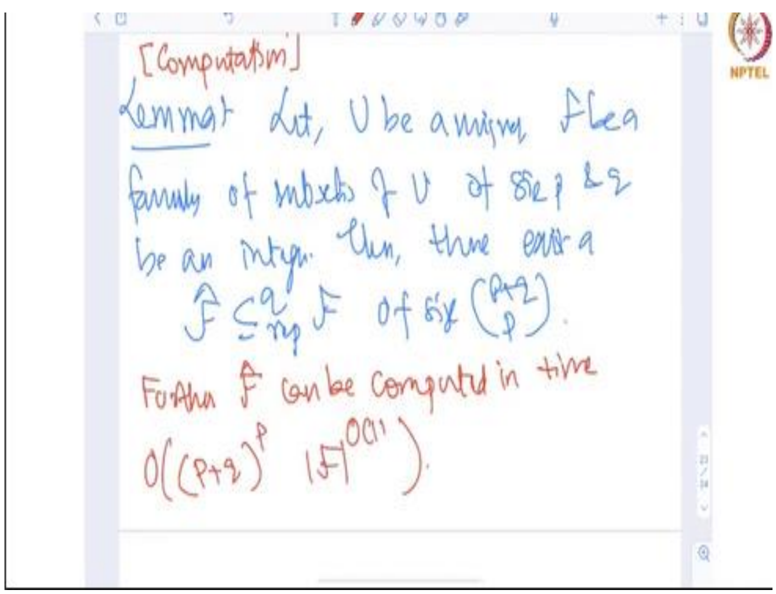
(Refer Slide Time: 21:48)



So, before we go about computational aspect of computing q representative family let us make a remark. So, what is the remark is two families theorem tight? And the answer is yes. Because you take universe equal to 1, 2, p + q and let us take family equal to all subsets of size p. Then for every x in F or rather every A i in F you take B i = U - A i. So, definitely A i and B i are disjoint and because of the cardinality constraint p + q every other thing will intersect with that.

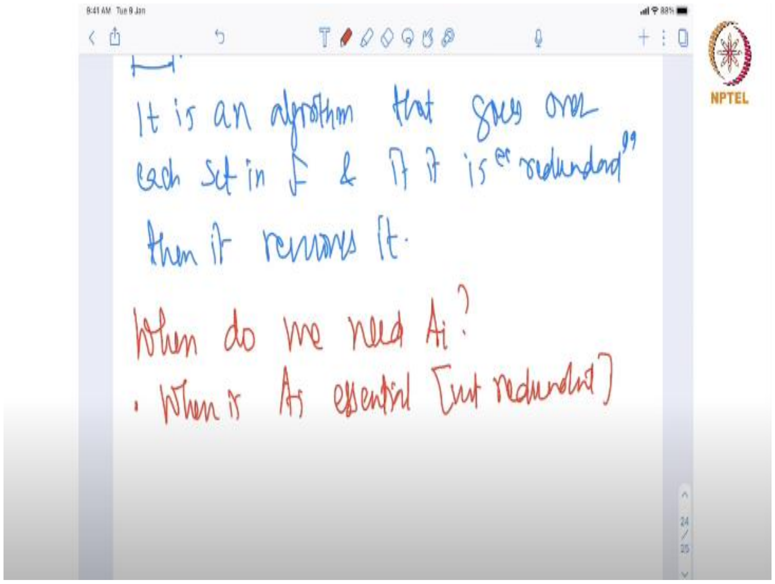
So, this shows that and since carnality of F is p + q choose p we can pair them. So, not only this is tight it also imply that in some sense representative family size is also tight. So, such an object exists..

(Refer Slide Time: 23:47)



So, this was existential. Now let us look at the computational lemma. This is computation. So, let U be a universe, F be a family of subsets of U of size p and q and be an integer then there exist. So, I would like to say further if F hat can be computed in time p + q. So, let us try to give a proof for this.

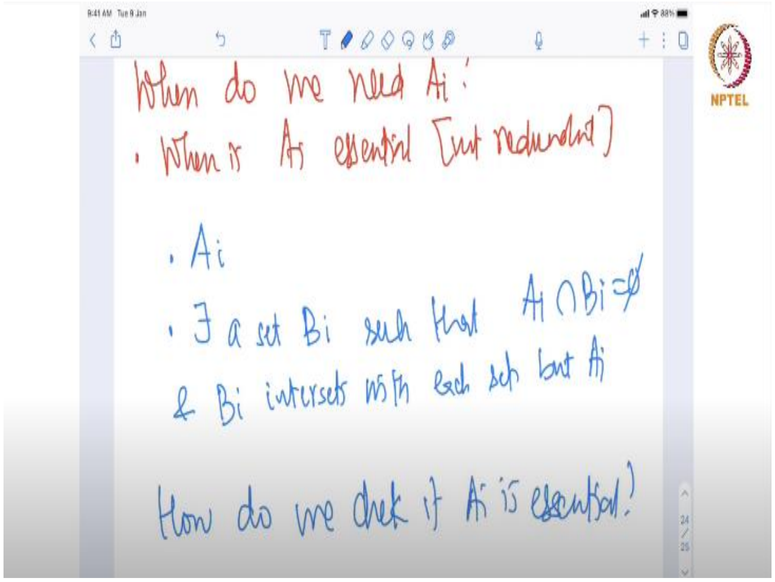
(Refer Slide Time: 24:55)



So, I am going to design an algorithm which will do this job for us. So, algorithm is very simple. It is an algorithm that goes over each set in F and if it is redundant then it removes it. So, this is an algorithm. So, what is this algorithm? It goes over each set in F and if it is redundant then it removes. And so, what are we trying to achieve? We are trying to get an F which is minimal. So, let us ask ourselves.

So, when do we need A i or rather let us say when do we need A i or rather when is A i essential? It is not redundant.

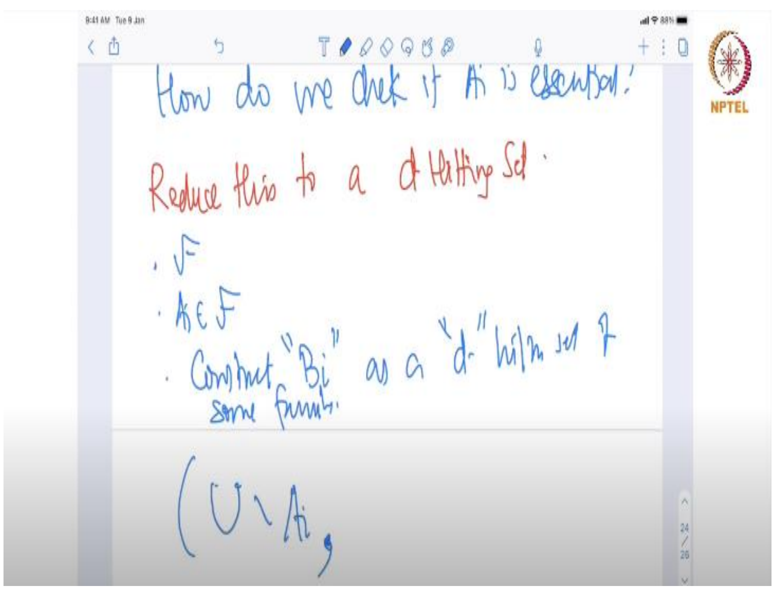
(Refer Slide Time: 26:37)



So, we already have seen that way to the definition of A i let us so here it is A i. It is necessary because there exist a set B i such that A i intersection B i is phi and so this should be not B i, A i intersection B i is phi and B i intersects with each set but A i. This is how we define the minimality. So, a set A i is essential if I can find a set B i. What is the property of that? Such that A i intersection B i is empty and B i intersects with each set but A i only then a set A i.

Because that is how we defined this. So, this is our computation. How do we check if A i is essential?

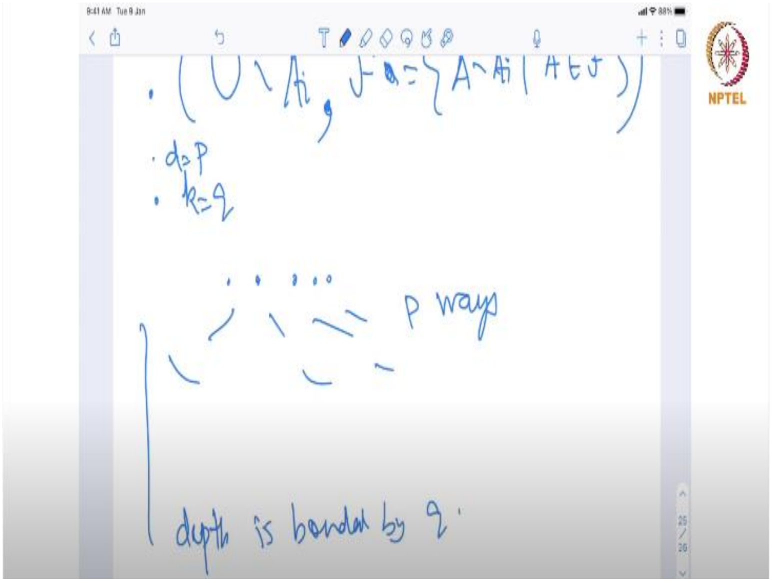
(Refer Slide Time: 28:11)



So, what we are going to do? Reduce this to a d hitting set problem. How? So, you are given a particular family F and A i is in F. So, now I am going to say look what is the meaning of A i essential? There exists a B i such that A i intersection B i is phi and B i intersects with each set but A i. Now what is the d hitting set or whatever the instance we are going to make you will see here. So, my instance is I am going to take a universe.

First of all, from here I remove A because you are not allowed to pick a set. Because the set which you want to pick. I am going to say construct B i as a d hitting set of some family. So, what is my family?

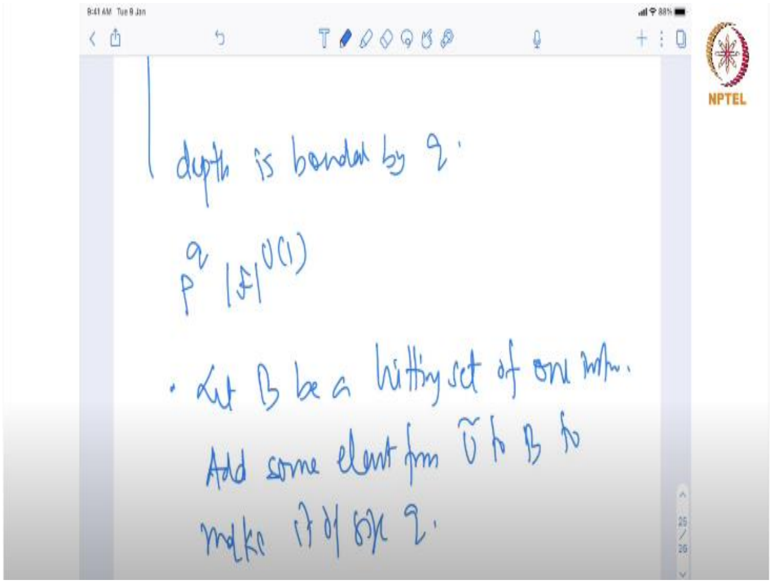
(Refer Slide Time: 29:35)



Some universe U - A i because B i has to be disjoint from A i and the family which I create is F tilde is nothing but A - A i A in my family. That is, it. And of course, remove empty set, there is nothing to hit, this is it. Now fix your parameter k is q, d is p. And if you do a branching, what is the branching algorithm? You pick up a set you know that you have to pick one of the element here. So, it is a very simple branching algorithm.

You pick up a p Xi set and you branch in p base and since you are looking for a solution of size q, the branching depth of this algorithm is bounded by q.

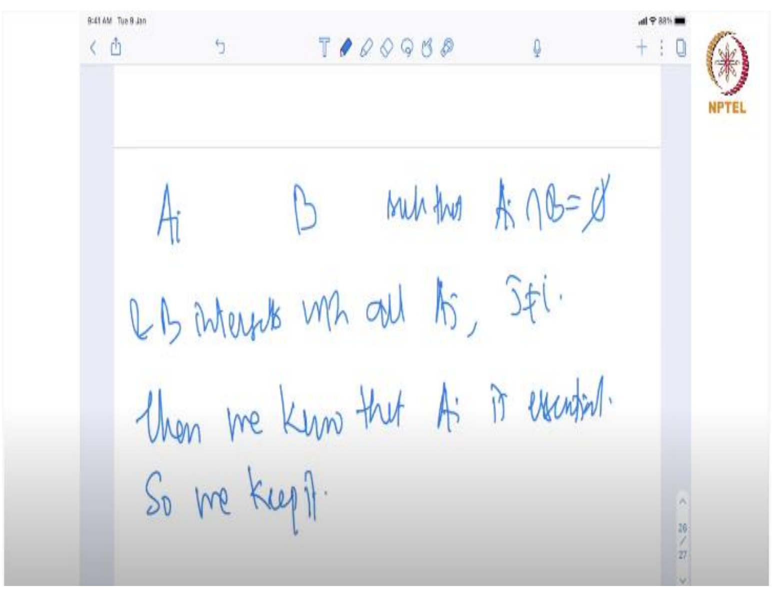
(Refer Slide Time: 30:47)



So, the running type of this algorithm is actually p to the power q something, some family because you have to check whether and take care of. So, basically what is B i? If you have a B i, if you have a set of size q, of course we have to assume that. So, now what can happen? You found a set U here. If you found a hitting set of size at most q for F tilde now you add. It could be of size at most q. So, add some redundant. Let B be a hitting set of our instance.

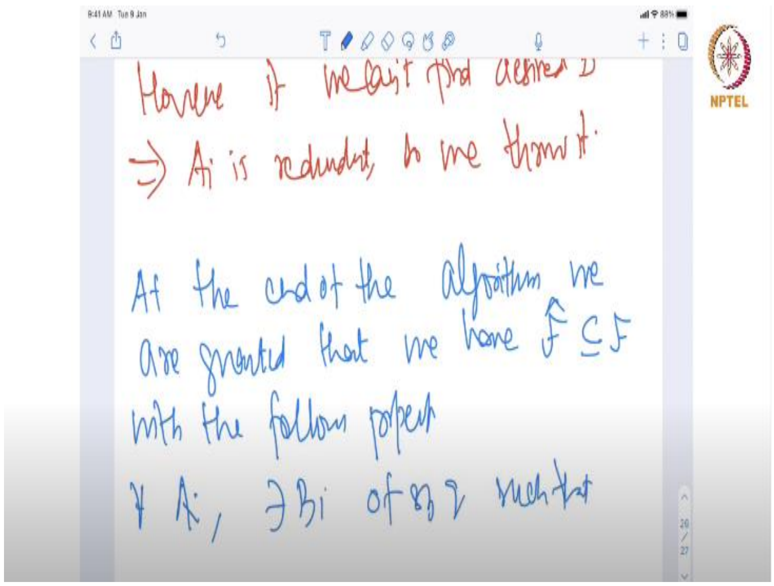
Then add some element from let us call it U tilde from U tilde to B to make it of size q. And we have such elements because A i had size p so even if you delete it B universe has size q. Now let B be a hitting set of one add some element. So, now what you know about this?

(Refer Slide Time: 32:40)



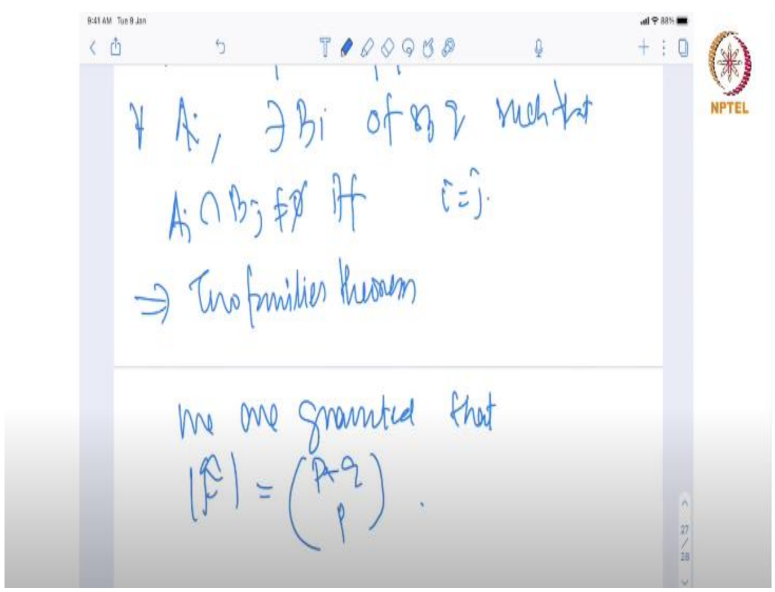
You have found for this A i a B such that A i intersection B is empty and B intersects with all A j, j not equal to i. Then what do you know? Then you know such an A i is essential. So, you keep such an i.

(Refer Slide Time: 33:30)



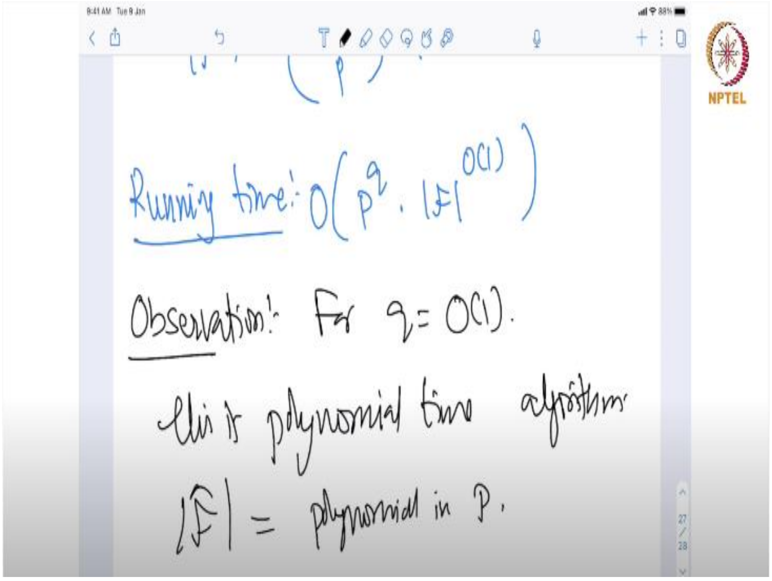
However, if we cannot find desired B what does it imply? This implies that A i is redundant. So, we throw it and notice that even after you throw the B which was guaranteeing that A i is disjoint from this B and B intersects everybody else that still holds. So, at the end of this algorithm we are guaranteed that we have an F hat subset of F with the following property. For each A i there exists a B i of size q.

(Refer Slide Time: 34:45)



Such that A i intersection B j is not equal to empty if and only if A i if and only if i = j. And now by two families theorem what are we guaranteed? We are guaranteed that.

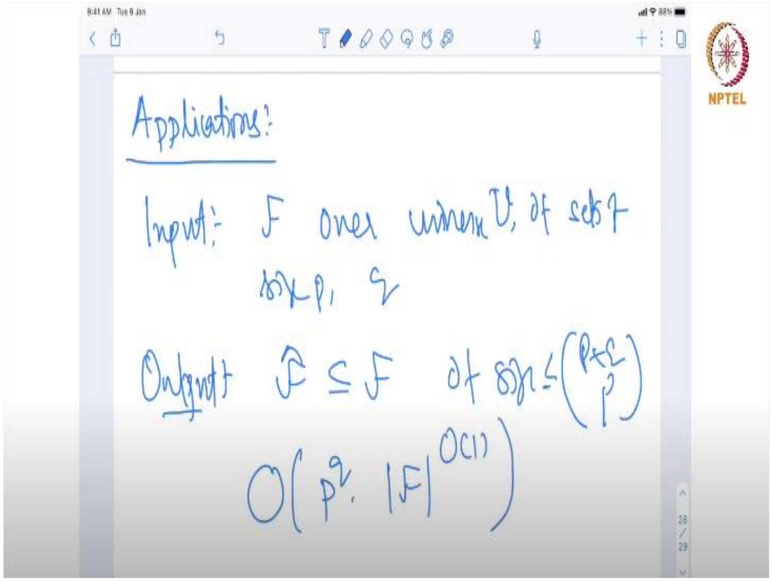
(Refer Slide Time: 35:35)



So, notice that we have found a family of optimum size but the running time is far from optimal in the sense that the running time of our algorithm is p to the power q. This is the running time. But some observations are in order. For q = order 1 this is a polynomial time algorithm. This is the first observation to make. So, when you are computing or trying to compute a representative with respect to like only constant size set from outside then actually you can get.

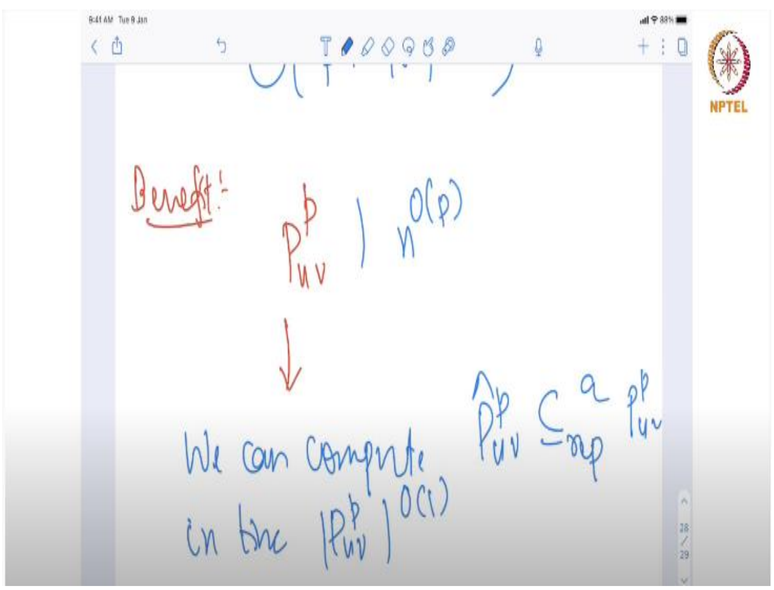
And whatever the size of F hat in this case will also be polynomial in B. This is another thing to worry about or to note. So, let us see how much we planned. What else did we plan to do now? We saw the bound on F hat, computation of F hat. Now we have to show how we could use these bounds and these algorithms to design algorithm. So, what is the running time of this algorithm? It says look if you give me a family then I can take running time polynomial in F and p to the power q to give you the set.

(Refer Slide Time: 37:48)



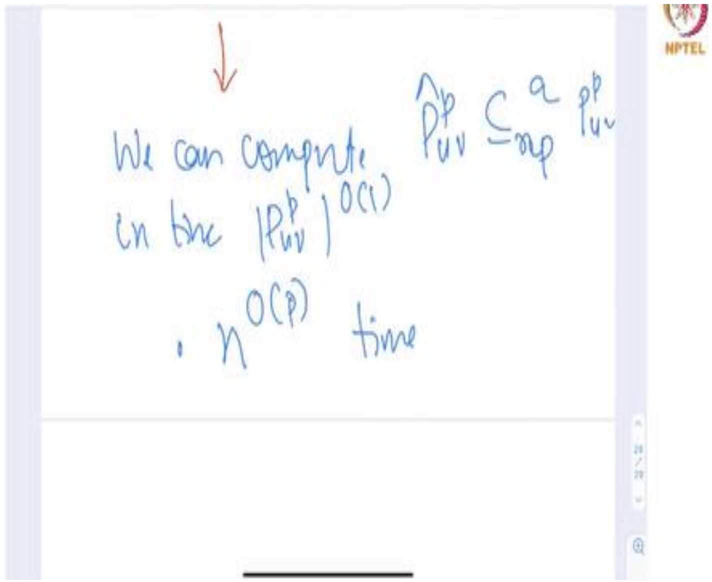
But imagine as we are told. So, now let us go back to applications. We will do applications now. So, imagine what this algorithm tells us? This algorithm tells us that look if you give me input and F over universe U of sets of size p and give me an integer q, I will output an F hat subset of F of size at most p + q choose p. But the running time of this algorithm is so first of all I mean what benefit are we going to get from this algorithm.

(Refer Slide Time: 39:00)



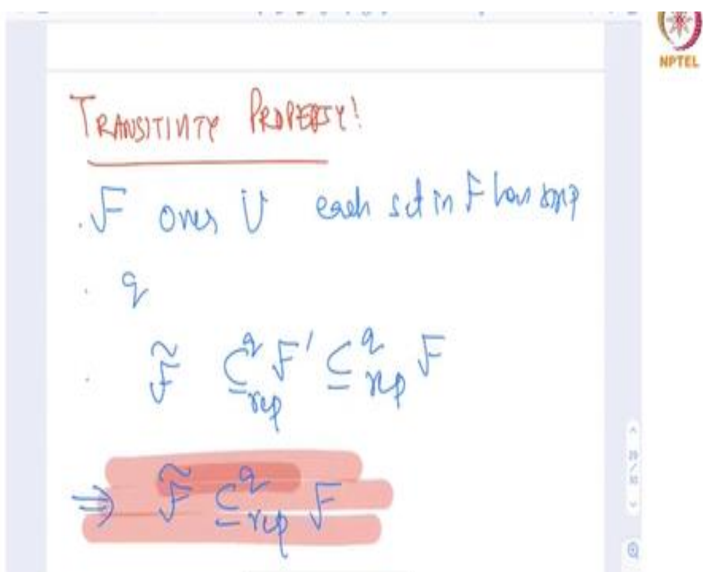
It says fine, if you give me say for example, for directed k path whatever we say look at st k, if you give me this, we can compute its representative let say q we can compute this. But in time how much? U v p size of this at least this much. Now we know that this size could be as bad as n to the power big of k.

(Refer Slide Time: 40:22)



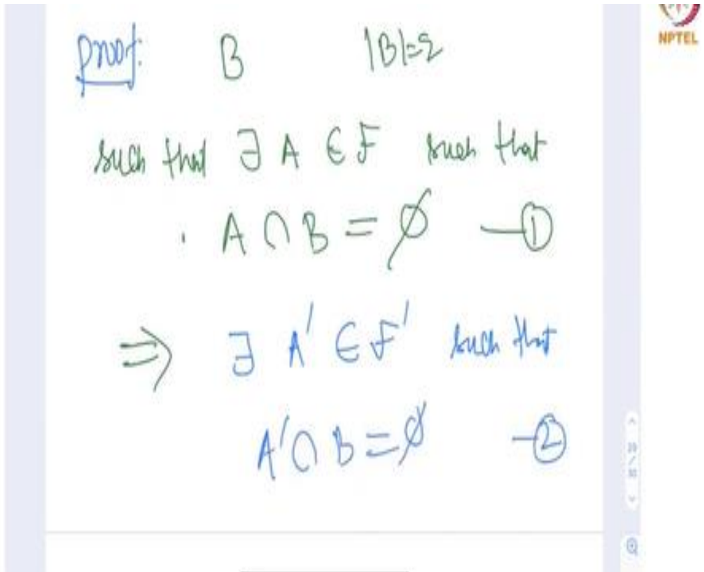
So, like this, computation access is going to take n to the power of p time. So, if I give you this itself explicit then what is the point of, I mean this computation overall. So, this does not make much sense in that regard. We will see how our dynamic programming can be applied in a very clean fashion that we will never have this whole set. But we still have a good enough set that if we compute a representative with respect to that we are still good.

(Refer Slide Time: 40:57)



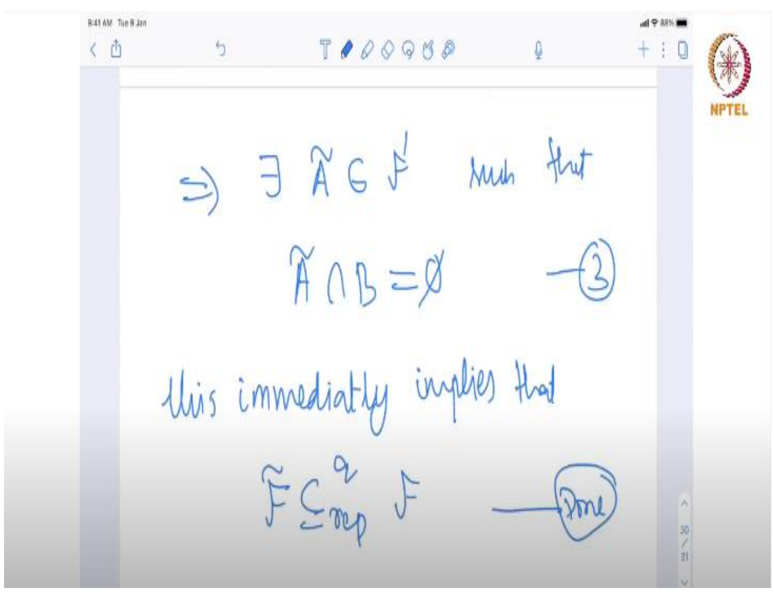
And for that the one property of this representative sets that are useful is the following transitivity property. So, what is that? So, suppose you are given a family F over U each set in F size p are given and you are given integer q. What I want to say is that look, suppose we have F prime which is say q representative of F and F tilde is q representative of F prime. This implies that F tilde is q representative of F. So, this is what I mean by transitivity and this is very easy to prove.

(Refer Slide Time: 42:16)



So, what is the first meaning of this F prime? We know that rather this is fine. So, now suppose you have a set you gave me let us say p such that there exists, A in F. Of course, it is phi, let us give this as 1. That is first. Now because of the first this implies there exists A prime in F prime such that A prime intersection B is phi. This is the second. Now F hat is a representative so now given that B and now you apply at A prime.

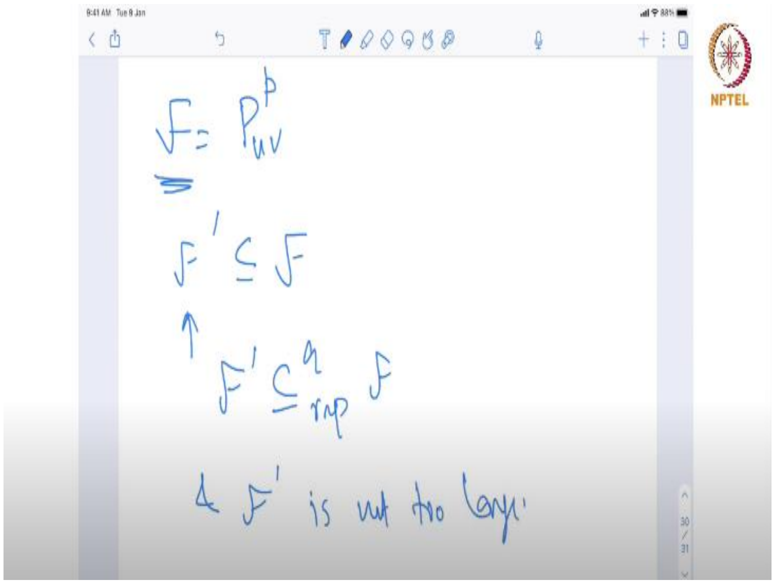
(Refer Slide Time: 43:40)



So, this implies that F hat is representative of F prime implies what? There exists A tilde in F prime such that A tilde intersection B is phi. So, what it means is that look given A, given B, if there exists an A prime. Now you use this in the same way, now you know that this A prime. Now because F hat is a representative of F prime means for every B if there is something in set F prime which it is disjoint with then F hat has to respect that.

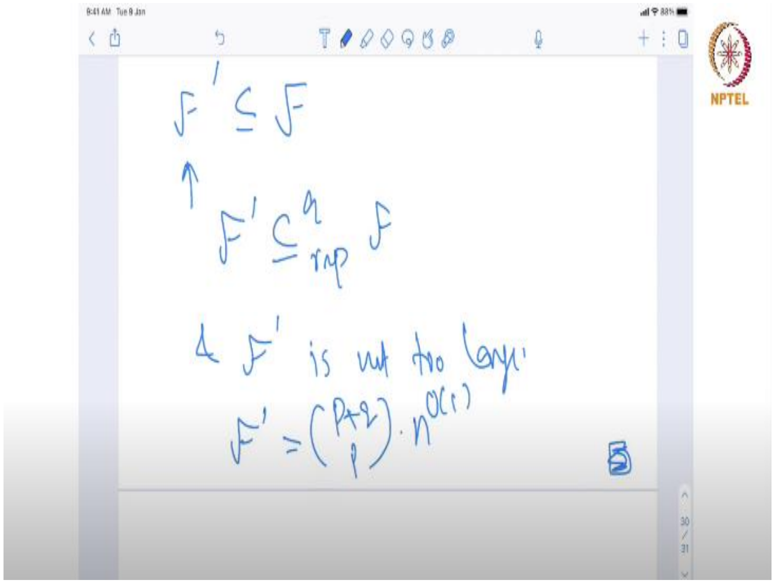
This implies that I can find A tilde. So, this immediately implies that F hat is a q representative of F, that is, it. So, why did I bring this?

(Refer Slide Time: 45:08)



Because look if the F we have to always work with is this. But we will never work with such an F. We will work with an F prime subset of F with the property that F prime is a q representative of F and F prime is not too large.

(Refer Slide Time: 45:46)



In fact, we will guarantee that F prime is at most some n to the power of 1. So, yes, it can be big but it is only big with respect to p + p q, slightly larger. And every time because we guarantee this, this is the F prime that will feed to our algorithm and get F hat. So, I think this is a good place to stop here and do the applications in the third lecture. So, in the first lecture we try to give motivation to this definition of representative families.

And how it is connected to two families theorem in the combinatorics, how we can use that combinatorial theorem to build up this machinery, how we can use just the normal knowledge of d hitting set algorithm to compute this to come with an algorithm or design an algorithm to compute this representative family. And in the next lecture we will show you how this transitivity could be used to design good efficient algorithms. So, with that we will close the second lecture.