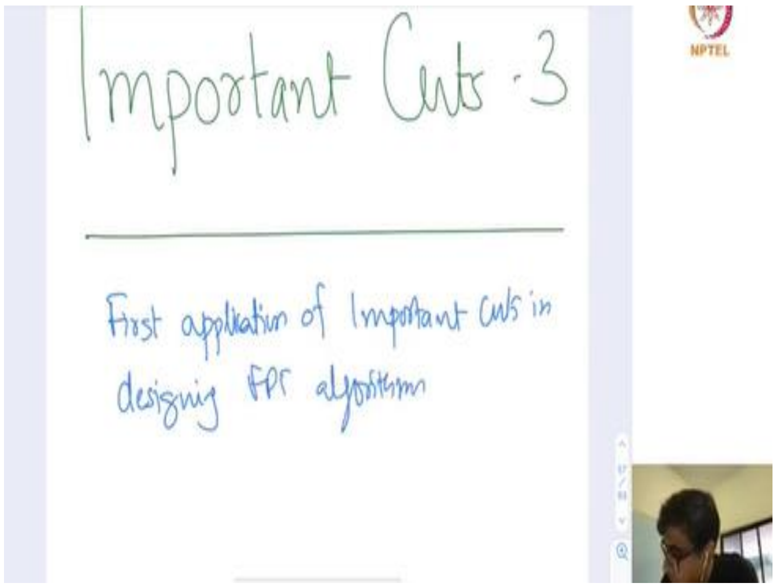
Parameterized Algorithms

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

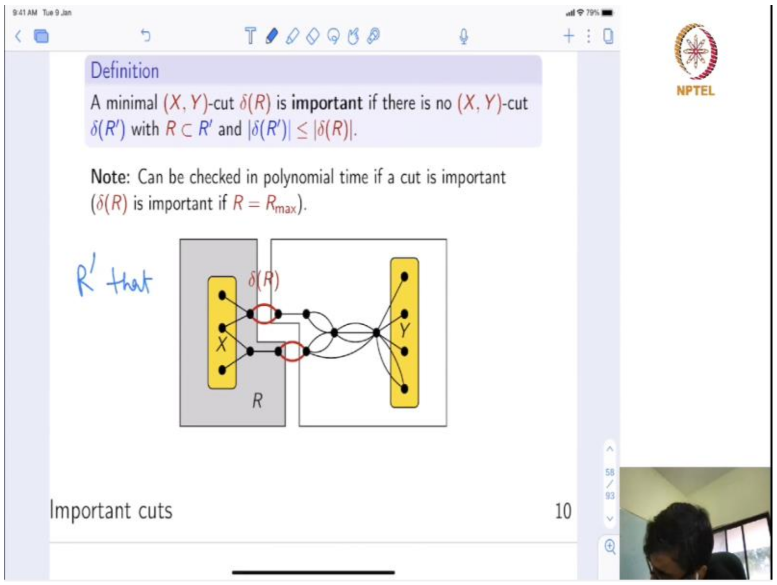
Lecture - 37
FPT Algorithm for Multiway Cut

(Refer Slide Time: 00:16)


Important Cuts · 3
First application of Important cuts in designing FPT algorithms
NPTEL

Welcome to the third lecture of on important cuts. In this lecture we will see our first application of important cuts in designing FPT algorithms. So, just let us just recall the definition.

(Refer Slide Time: 00:43)



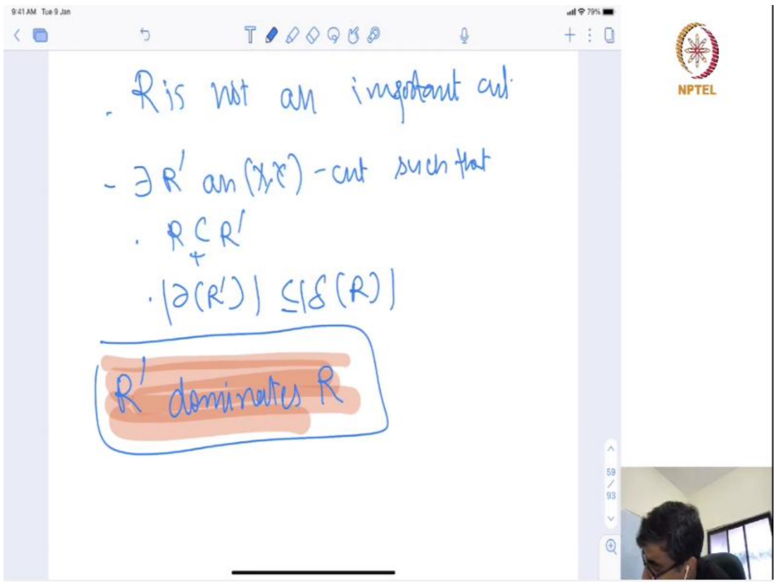
Definition

A minimal (X, Y)-cut $\delta(R)$ is **important** if there is no (X, Y)-cut $\delta(R')$ with $R \subset R'$ and $|\delta(R')| \leq |\delta(R)|$.

Note: Can be checked in polynomial time if a cut is important ($\delta(R)$ is important if $R = R_{max}$).

If you recall correctly or minimal X, Y or delta R is important, if there is no X, Y delta R prime with R subset of R proper subset of R prime. And the number of edges between this vertical delta R prime is at most delta these are called important curves. And if you have if you and R prime that vertical R prime that.

(Refer Slide Time: 01:22)

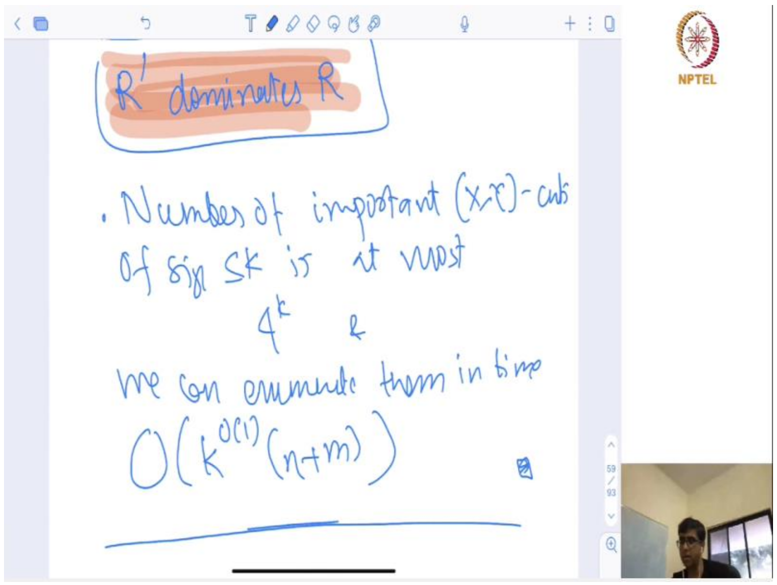


- R is not an important cut
- $\exists R'$ an (X,Y)-cut such that
  - $R \subsetneq R'$
  - $|\partial(R')| \leq |\delta(R)|$

$\boxed{R' \text{ dominates } R}$

So, let us fix a terminology just to help ourselves. So, suppose R is not an important cut. Then we know that exists R prime and X, Y cut such that R is a proper subset of R prime and delta R prime this. We will call R prime we will say R prime dominates that. So, we will just use this terminology for which will be very helpful. And what we saw last night that.

(Refer Slide Time: 02:31)



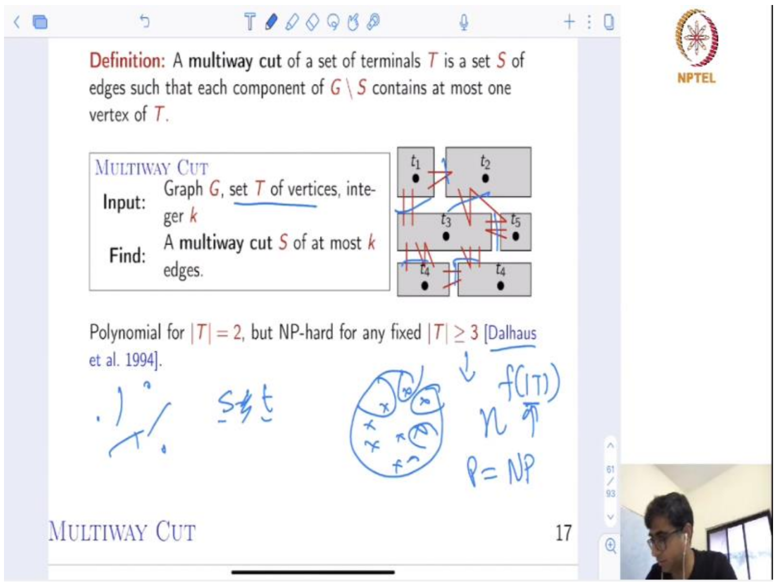
$\boxed{R' \text{ dominates } R}$

- Number of important (X,Y)-cuts of size $\leq k$ is at most 4^k &

we can enumerate them in time $O(k^{O(1)}(n+m))$

So, this is the fact we learned a number of important X, Y cuts have size at most k is at most 4 power k and we can enumerate them in time big O of k to the power big O of 1 n + m. And so, we saw that algorithm in this fashion. Now, we are going to see the first application of this integration algorithm.

(Refer Slide Time: 03:19)



Definition: A multiway cut of a set of terminals T is a set S of edges such that each component of $G \setminus S$ contains at most one vertex of T.

MULTIWAY CUT	
Input:	Graph G, set T of vertices, integer k
Find:	A multiway cut S of at most k edges.

Polynomial for $|T| = 2$, but NP-hard for any fixed $|T| \geq 3$ [Dalhaus et al. 1994].

And so, the first problem we are going to talk about is multiway cut. So, what is a multiway cut a multi bake cut of a set of terminals T is a set S of edges such that each component of G - S contains at most one vertex. So, s-t cut is like you want to delete minimum number of edges so, that s and t are in different components, but now we do not have 1 terminal we have set of terminals and our objective is to delete at most k edges.

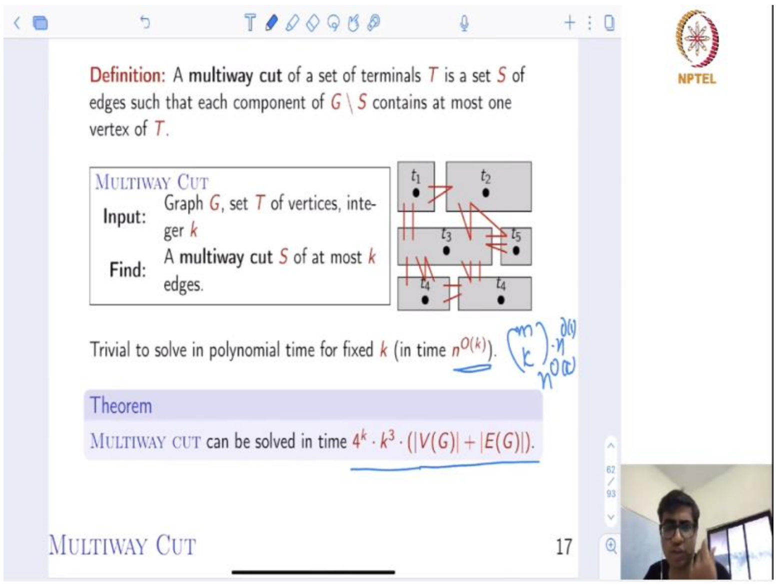
So, that every terminal is in single component it means we have been able to disconnect parts between every pair t i t j in so, for example look at this example. Now, if you delete all these red edges then each of these terminal are in their own components. They are in other words, there are no 2 vertices in my terminal t such a way that the part between them after we get deleted the s edges.

So, what is a multiway cut problem is a multiway cut problem if you are given a graph G a set T of terminals and integer k and our objective is to check whether there exists a set S of utmost k multiway cut is of utmost k edges. And what is a multiway cut? A set of edges whose deletion; results in every terminal being in a single component or every component containing at most one terminal.

So, this is definitely polynomial time for cardinal T = 2 but it is actually NP hard even if you had just 3 terminal and he wanted to delete minimum number of edges then this problem is known to be NP hard and it is known from by paper by Dell house from 1990 itself. What is this implies? That implies this NP hardness reduction that you cannot have an algorithm of running time even n to the power F of terminal because this will imply P equal to.

So, we do not hope to have an (()) (05:22). So, number of terminal is not an important parameter for this problem in general. So, the parameter which we will use is the solution size number of edges that we need to delete such that these terminals get separated from each.

(Refer Slide Time: 05:37)



Definition: A **multiway cut** of a set of terminals T is a set S of edges such that each component of $G \setminus S$ contains at most one vertex of T.

Multiway Cut

Input: Graph G, set T of vertices, integer k

Find: A **multiway cut** S of at most k edges.

Trivial to solve in polynomial time for fixed k (in time $n^{O(k)}$). $\binom{m}{k} \cdot n^{O(1)}$ $n^{O(k)}$

Theorem

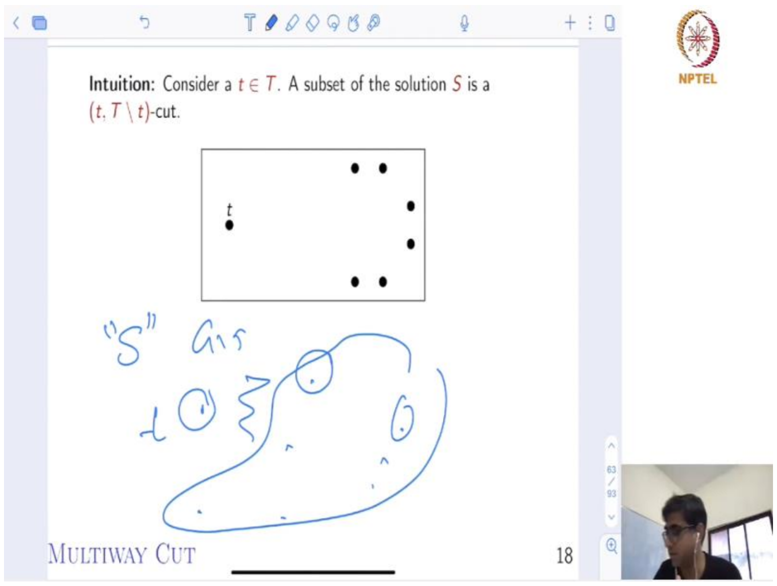
Multiway cut can be solved in time $4^k \cdot k^3 \cdot (|V(G)| + |E(G)|)$.

Multiway Cut 17

Clearly it is trivial to solve this problem in polynomial time for every fixed gate you try all possible cases subset and check whether each terminals belongs to like or each component contains at least one terminal. And this can be done in time into the code because like you try m choose k number of edges and then you do in this. So, this will lead to enter but what we are going to show to you that multiway can actually can be solved in time 4 to the power k, k to the power q V G + E G.

We will start with giving us giving you an algorithm which has this running time, but we will first analyse this slightly easily and then refine our analysis to achieve this kind of effect I hope this is clear. So, just you can take you can pause for a few seconds at this point of time and phrases definitions properly and then move into that.

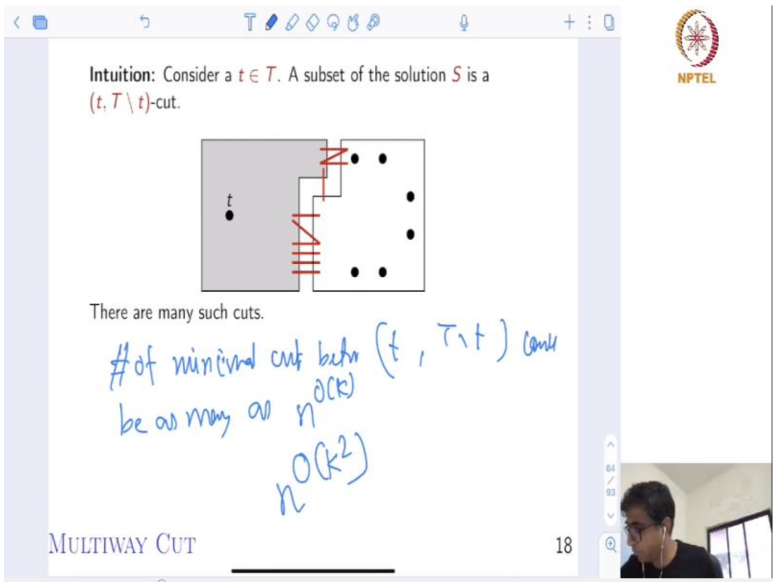
(Refer Slide Time: 06:36)


Intuition: Consider a $t \in T$. A subset of the solution S is a $(t, T \setminus t)$-cut.

So, let us consider the intuition suppose t is T and a subset of the solution S is a t, T manner side. Look what property that? Look at S if you are going to delete G - S then every terminal is in single like every terminal is in its own component. Then there will be some subset of edges which deletion will disconnect t from everybody else. In particular the whole set S itself does the job.

So, there could be a minimal subset of S which does the job of the protocol reducing to from every other.

(Refer Slide Time: 07:18)


Intuition: Consider a $t \in T$. A subset of the solution S is a $(t, T \setminus t)$-cut.

There are many such cuts.

# of minimal cut betw $(t, T \setminus t)$ could be as many as $n^{O(k)}$

$n^{O(k^2)}$

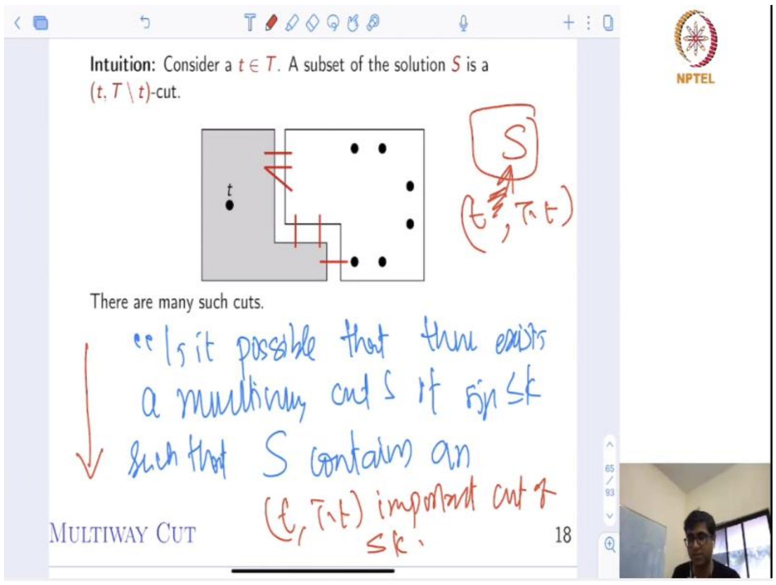
And so, a subset of the solution S is t, T - t, but there are many such cuts I mean if you just wanted to disconnect you know, if we know that our protocol that we know that the number of min cuts or minimal cuts between t and T - t could be as many as n to the power of k. So, an algorithm where you would like to enumerate all minimal cuts, delete it and then try to disconnect everybody else will not lead to a good role with that.

Because that; could just become n to the power of big O k 2. So, that is not good. We know that my optimum solution contains a minimum solution with minimum cut between t and T - t. So, a natural algorithm will be enumerate a minimal cut between t and T - 8 of size at mostly delete this and in the recursively try to solve disconnect every other. Because after this you know that maybe some set of components some set of terminals have gone into component.

Some set of terminals still need to separate it. So, you pick one such terminals try to separate it from the other you do this and every time you know that you are going to incur at least cost of one. So, this branching tree is going to be at most k, but the branching depth is going to be n to the power big of k because you are eliminating all minimal costs. So, there are many such costs. So, this is and good this is a natural algorithm.

But this does not work because number of minimal cuts as we have seen could be as bad as into the power big of (()) (09:00). So, now rather than doing this we ask ourselves what happens if we enumerate t T - t important cuts because we know they are only 4 to the power k, so naturally at this point of time after developing all this.

(Refer Slide Time: 09:16)



Intuition: Consider a $t \in T$. A subset of the solution S is a $(t, T \setminus t)$-cut.

There are many such cuts.

"Is it possible that there exists a multiway cut S of size $\leq k$ such that S contains an $(t, T\setminus t)$ important cut of size $\leq k$.

MULTIWAY CUT 18

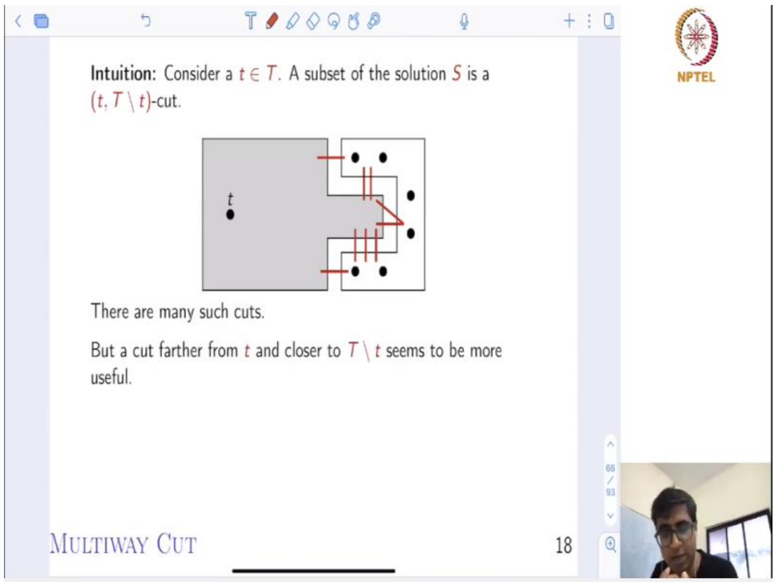
All this can we ask ourselves is it possible that there exist a multiway cut S of size at most k such that S contains an t T - t important cut of size at most k. So, we know that a minimal cut, a minimal s-t, t T - t cut is contained inside is that is. But now I am saying look that but that is not useful because there are too many of them, I am saying can I find a solution is to my problem, which has a property that if I look at t T - t solution inside this.

This is not any minimal cut, but this is an important cut. Because if we had this, then we know that we can enumerate each of these cuts delete a solution and in the remaining try to separate those pair of terminals which are not yet separated. We know that every time we have to pay something. So, that depth will decrease by one each time and since the branching factor is upper bounded by 4 to the power k. This will lead to a 4 to the power k squared algorithm.

But still, this will be an FPT algorithm. So, that is simple. So, our main objective will be is to show you a statement is true. And in fact, what we will be able to show that actually such a statement is true. And that is the crux of any cut kind of algorithm that we are trying to show that look, we can separate one guy from everybody else. And there are some signals like there are too many minimal cuts.

But there are some small number of cuts which we will call important cuts which is good enough for us to eliminate we do not need to enumerate all of these cuts, because for any minimal cut, if you look at an important cut, that dominates it, that does the same job as this minimal cut will do. And that is the crux of this lemma.

(Refer Slide Time: 11:38)



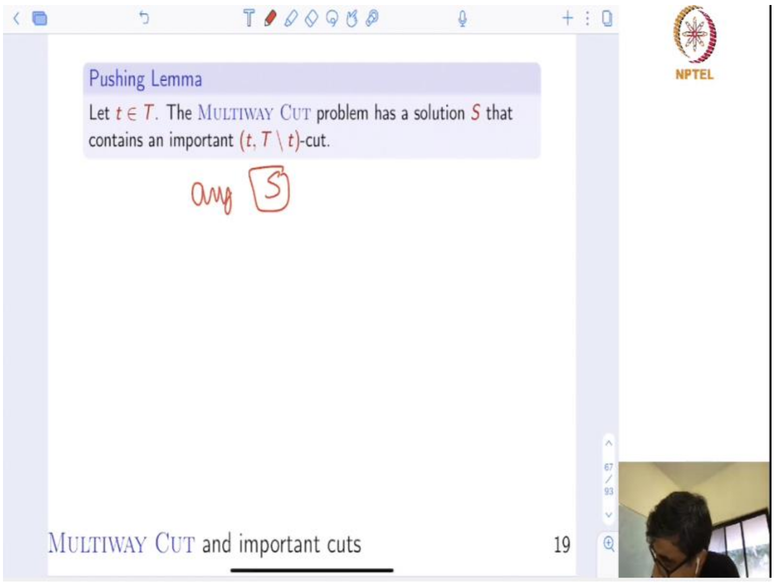
Intuition: Consider a $t \in T$. A subset of the solution S is a $(t, T \setminus t)$-cut.

There are many such cuts.

But a cut farther from t and closer to $T \setminus t$ seems to be more useful.

And these kinds of lemma are called what is that called? A pushing lemma. And this is what the intuition is trying to do that occurred farther from t and closure to T - t seems to be more useful than any minimal cut.

(Refer Slide Time: 11:52)



Pushing Lemma

Let $t \in T$. The MULTIWAY CUT problem has a solution S that contains an important $(t, T \setminus t)$-cut.

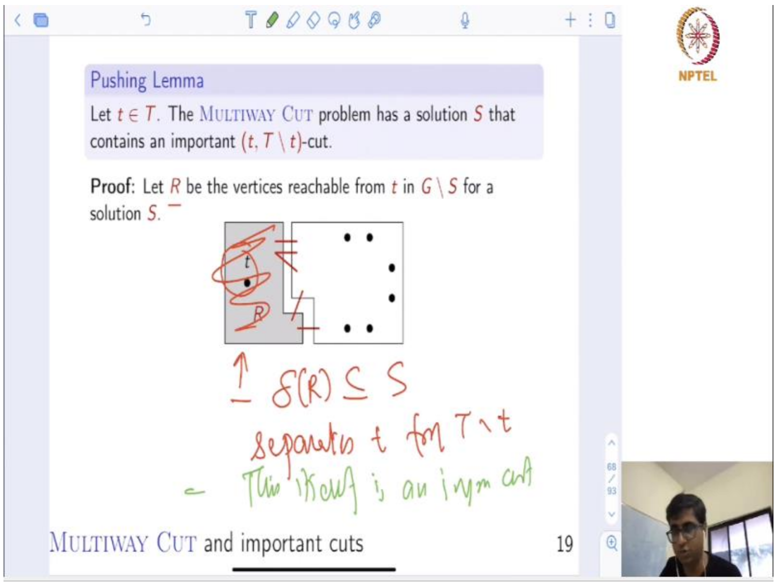
any $\boxed{S}$

MULTIWAY CUT and important cuts 19

And this is the intuition, which is what is that call which is encapsulated in the pushing lemma like we can push a solution we can push a solution. Well, one of the important because what is your property important cut is like look at this important cut, I cannot push this further without violating either the solution constraint the size of the solution constraint, or the fact that R is not contained.

So, this is what we are going to show that basically we are going to show to this pushing lemma has a solution is that contains that importantly demands to this is proof before we go to prove heritage, the proof is very simple, so you fix any S.

(Refer Slide Time: 12:40)



Pushing Lemma

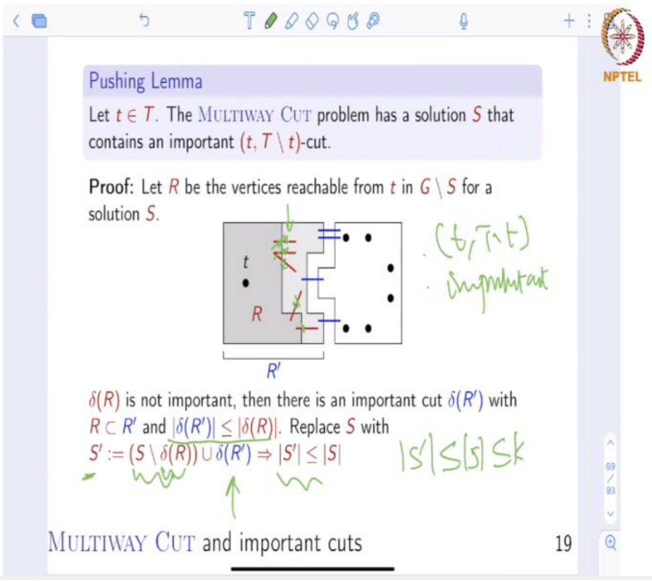
Let $t \in T$. The Multiway Cut problem has a solution S that contains an important $(t, T \setminus t)$-cut.

Proof: Let R be the vertices reachable from t in $G \setminus S$ for a solution S.

Multiway Cut and important cuts 19

And let R be the set of vertices reachable from G - S for a solution S. What is R? R is a set of vertices which is reachable from t. So, these are the vertices which I can read. So, this is my first so notice that delta R which is a subset of S separates t from T – t. This means, there are 2 case edges this itself is an important then we are done, is that particular solution which has this property.

(Refer Slide Time: 13:33)



Pushing Lemma

Let $t \in T$. The Multiway Cut problem has a solution S that contains an important $(t, T \setminus t)$-cut.

Proof: Let R be the vertices reachable from t in $G \setminus S$ for a solution S.

$\delta(R)$ is not important, then there is an important cut $\delta(R')$ with $R \subset R'$ and $|\delta(R')| \leq |\delta(R)|$. Replace S with $S' := (S \setminus \delta(R)) \cup \delta(R') \Rightarrow |S'| \leq |S|$

$|S'| \leq |S| \leq k$

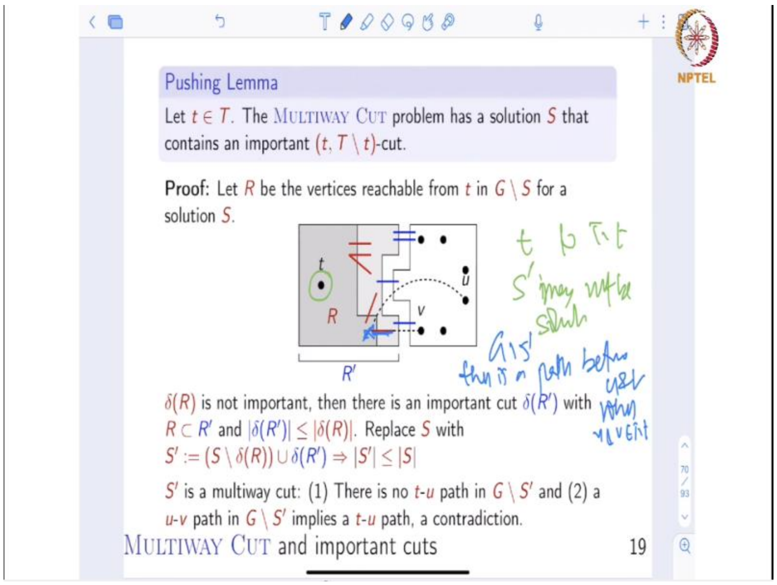
Multiway Cut and important cuts 19

Otherwise, you know, if delta R is not an important cut then there exists some R prime. So, look at this blue, so red edges are so, this is the edges, these are the edges of R, but I know is not an important cut. So, what I am going to do? I am going to let our prime is t T - t important cut R prime. And what are the property of this? So, the number of blue edges is less than the red edges, which is encapsulated by the delta R prime is at most delta R.

So, now I am going to make a new solution, what I am going to make a solution? I said from I am going to make a new solution is prime from S I am going to remove all the red edges, the edges, which was incident to R rate delta R, and from there I am going to add delta R. Now, what look I deleted first delta R and then I added delta R prime the size of delta R prime is at most delta prime, which implies that the size wise S prime is at most f which is at most case, which is great.

But now we have to show that S prime is also a multiway cut. Meaning there is no path between t i and t j between for any set of pair of terminals.

(Refer Slide Time: 15:00)



Pushing Lemma

Let $t \in T$. The Multiway Cut problem has a solution S that contains an important $(t, T \setminus t)$-cut.

Proof: Let R be the vertices reachable from t in $G \setminus S$ for a solution S.

$\delta(R)$ is not important, then there is an important cut $\delta(R')$ with $R \subset R'$ and $|\delta(R')| \leq |\delta(R)|$. Replace S with $S' := (S \setminus \delta(R)) \cup \delta(R') \Rightarrow |S'| \leq |S|$

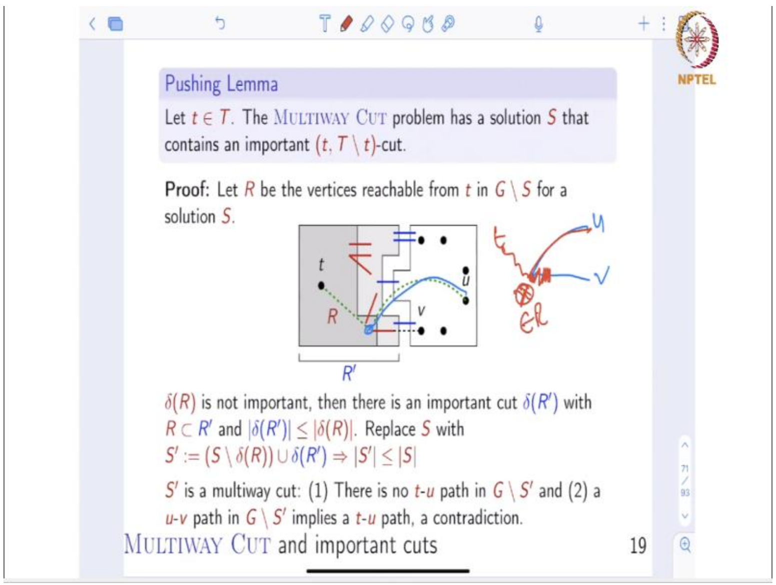
S' is a multiway cut: (1) There is no t-u path in $G \setminus S'$ and (2) a u-v path in $G \setminus S'$ implies a t-u path, a contradiction.

Multiway Cut and important cuts 19

First thing first, notice there cannot be a path from T and another there cannot be a path between t to some T - t some u or v. Why? Because R prime delta R prime is an important t T – t cut. So, you are still separating this so the only reason you could S prime may not be a solution. Because after in because in G - S prime what happens because in G - S prime there is a path between u and v where u and v belongs to the T - t.

But I know that there was no path before, so how did suddenly I get a path from u and v well the reason you can get any path from u and v is because this path u and v must be containing one of the red edges. But now notice that your red edges then you must be coming inside because look to use this red edge you must be coming inside R and then going back to me. But then what implies? What was the property of the set R? The property the; set R that every vertex was reachable from T.

(Refer Slide Time: 16:34)



Pushing Lemma

Let $t \in T$. The MULTIWAY CUT problem has a solution S that contains an important $(t, T \setminus t)$-cut.

Proof: Let R be the vertices reachable from t in $G \setminus S$ for a solution S.

$\delta(R)$ is not important, then there is an important cut $\delta(R')$ with $R \subset R'$ and $|\delta(R')| \leq |\delta(R)|$. Replace S with $S' := (S \setminus \delta(R)) \cup \delta(R') \Rightarrow |S'| \leq |S|$

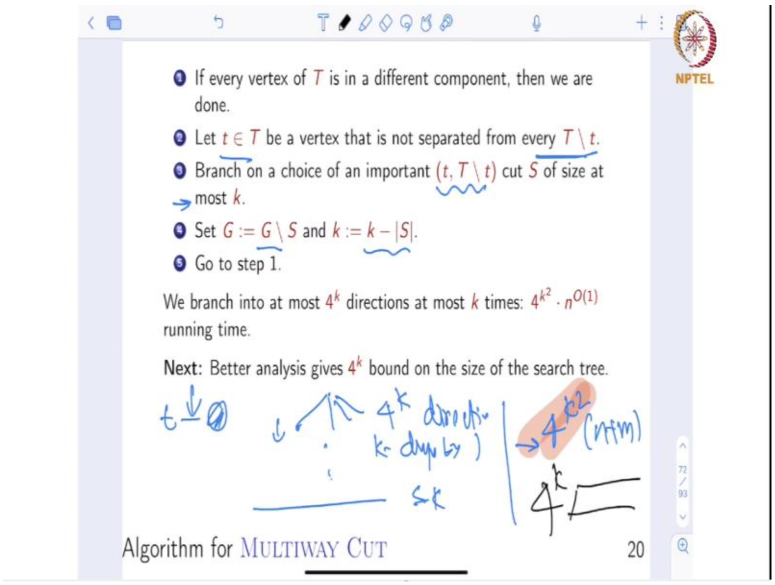
S' is a multiway cut: (1) There is no t-u path in $G \setminus S'$ and (2) a u-v path in $G \setminus S'$ implies a t-u path, a contradiction.

MULTIWAY CUT and important cuts 19

Which implies that I can reach t to these vertices and from there I could go to you, but that is a contradiction. Because look what did I do so, there is a path from u to v since this has to use a red edge otherwise there cannot be. If it uses red edge, it like one of the red edge there is a vertex here which is inside R now you know that every vertex is reachable from, so you come to T to x and from T to this vertex x two eight there is a path from v to u.

But u told me that he cannot reach any vertex in T - t because R prime was T - t separate. It is a contradiction. So, what are we able to show it is a very simple pushing lemma which tells us that there exists a solution S which contains an important cut between t and T - t and once you have shown this lemma your algorithm is more or less done.

(Refer Slide Time: 17:37)



1. If every vertex of T is in a different component, then we are done.
2. Let $t \in T$ be a vertex that is not separated from every $T \setminus t$.
3. Branch on a choice of an important $(t, T \setminus t)$ cut S of size at most k.
4. Set $G := G \setminus S$ and $k := k - |S|$.
5. Go to step 1.

We branch into at most 4^k directions at most k times: $4^{k^2} \cdot n^{O(1)}$ running time.

Next: Better analysis gives 4^k bound on the size of the search tree.

Algorithm for MULTIWAY CUT 20

So, here is an algorithm if every vertex of tuition different component when we are done. Otherwise, you choose a vertex t in T you choose a vertex t in T that is not separated from every T – t, if there is some vertex that is apart. So, you branch on a choice of importance T - t cut S size of at most k. You set G, G - S and you drop k by k - S and you go to step 1. So, what is your branching step every time you branch you branch into 4 to the power k directions.

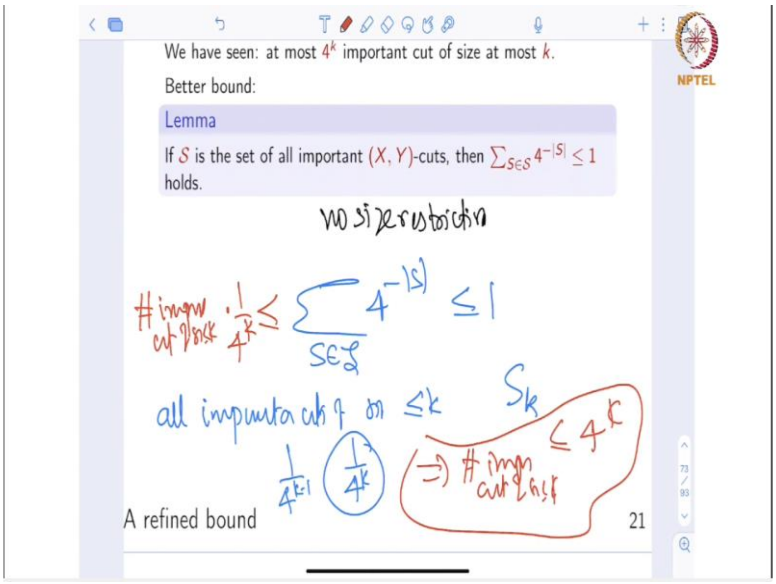
And every time why are you branching out? Branching because you want to separate t from something. So, you at least need one is to separate t from T - t because there is at least one part which implies that in the each branch k drops by 1. So, the branching depth of this tree is upper bounded by k. So, the total number of is like 4 to the power k. So, what did we get? So, this tells us that the number of this tells us that algorithm has running times.

So, to the power k square n to the power big O of 1 or in fact this is like. So, what is an algorithm? You rather than enumerating minimal cuts you pick up vertex D which still needs to be separated. Enumerate on the minimal cuts you are guaranteed that there exists an optimum solution containing one of the important cuts you find this important like you branches through this important cut and you delete this set S and then in the remaining you try to cut whatever needs to be cut.

And you know that because of my pushing lemma, you know that you are there is one branch where you are making a correct choice. And hence, you this will lead to an algorithm. And now each branching each branching factor is like the 4 to the power k because you are branching 4 to the power k direction, the depth of this branching is k. So, the total number of nodes in this branching t is upper bounded by 4 to the power k square.

So, that automatically bounds the running time by 4 to the power k squared poly or n to the power big of 1. You do not want to but we promise that we will give you an algorithm which is whose running time is actually 4 to the power k and some dependence on poly k and this is too far. So, for that we need a better analysis.

(Refer Slide Time: 20:16)



We have seen: at most 4^k important cut of size at most k.

Better bound:

Lemma

If S is the set of all important (X, Y)-cuts, then $\sum_{S \in \mathcal{S}} 4^{-|S|} \leq 1$ holds.

no size restriction

$$\#\text{impo cut of size} \leq k \cdot \frac{1}{4^k} \leq \sum_{S \in \mathcal{S}} 4^{-|S|} \leq 1$$

all important cuts of size $\leq k$ $\quad S_k$

$\frac{1}{4^{k-1}} \quad \boxed{\frac{1}{4^k}} \quad \Rightarrow \#\text{impo cut} \leq k \leq 4^k$

A refined bound

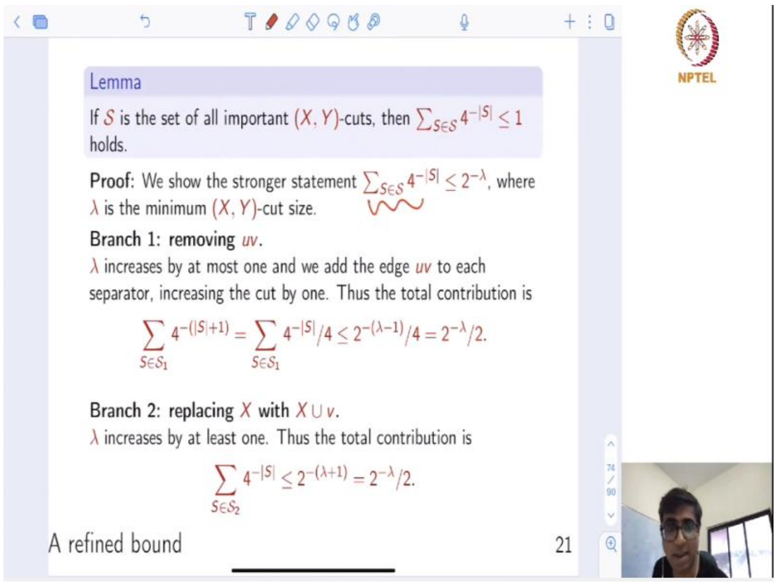
21

Let us see why this implies? If S is the set of all important X, Y cuts. We have seen at most 4 to the power k important cuts of size that will give a better bound edge. If S is the set of all important X, Y cuts all important no size restriction. Then what we know what do they say look at S in curly S are the set of all important separator 4 to the power - S is at most S. But you may ask why do I say this is not much more refined bound?

Well, look at all important cuts of size or at least k, all important cuts. So, let us call this S k, look this is like so 1 to the power 4 to the power k is smaller than 1 to the power 4 to the power k – 1. So, I can come up with a lower bound, this is less than equal to number of important cuts of size at most k times 1 to the power 4 to the power like this. So, that implies that number of important cuts of size at most k is upper bounded by 4 to power.

So, this is why I am going to call you that this is a refined one what we are saying that number of if you some that if you look at the contributions of 1 by 4 to the power important cuts as the size increases their contribution to this space of important cuts decreases in with respect to this and this is already too.

(Refer Slide Time: 22:06)



Lemma

If S is the set of all important (X, Y)-cuts, then $\sum_{S\in\mathcal{S}} 4^{-|S|} \leq 1$ holds.

Proof: We show the stronger statement $\sum_{S\in\mathcal{S}} 4^{-|S|} \leq 2^{-\lambda}$, where λ is the minimum (X, Y)-cut size.

Branch 1: removing uv.

λ increases by at most one and we add the edge uv to each separator, increasing the cut by one. Thus the total contribution is

$$\sum_{S\in\mathcal{S}_1} 4^{-(|S|+1)} = \sum_{S\in\mathcal{S}_1} 4^{-|S|}/4 \leq 2^{-(\lambda-1)}/4 = 2^{-\lambda}/2.$$

Branch 2: replacing X with $X \cup v$.

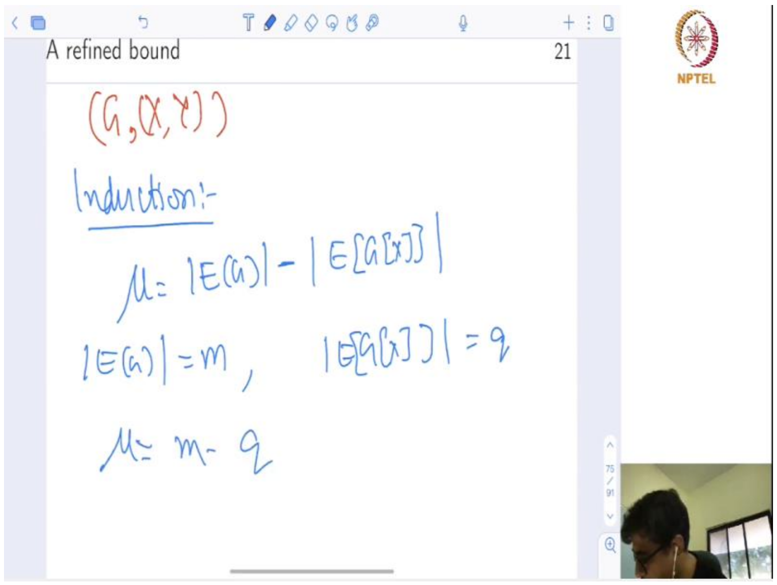
λ increases by at least one. Thus the total contribution is

$$\sum_{S\in\mathcal{S}_2} 4^{-|S|} \leq 2^{-(\lambda+1)} = 2^{-\lambda}/2.$$

A refined bound 21

In fact, what we will show is that number of important input like 4 to the power - S where S is belonging to report and separated is 2 to the power - lambda where lambda is the minimum X, Y cuts. And the way we will show this is by induction and induction will be on slightly different object let me show you on what.

(Refer Slide Time: 22:33)



A refined bound 21

$(G, (X, Y))$

Induction:-

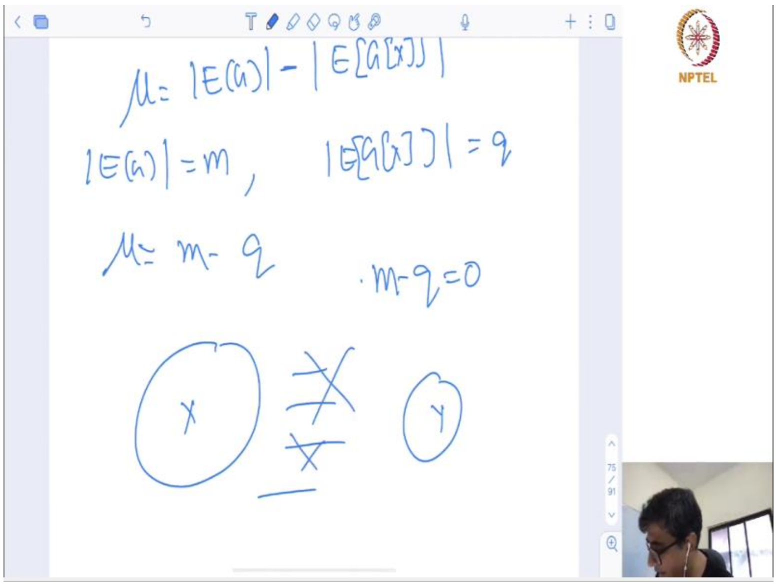
$\mu = |E(G)| - |E[G[X]]|$

$|E(G)| = m, \quad |E[G[X]]| = q$

$\mu = m - q$

So, given a graph G X and Y so; induction is going to be measured where it is going to be number of edges in my graph - number of edges in the graph induced on x. So, this is what my major is going to be. So, suppose I denote number of edges by m and number of edges in the graph induced on x to be let us say q. Then my major is going to be m is going to be m - q.

(Refer Slide Time: 23:32)



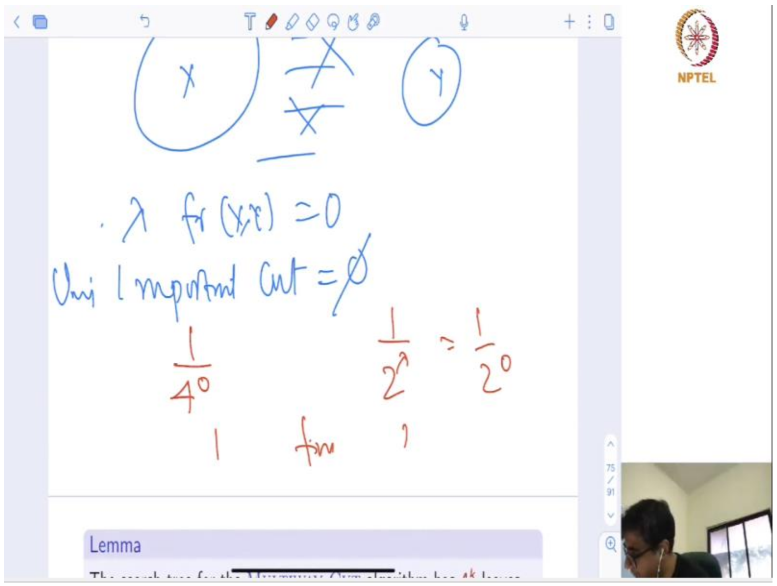
$\mu = |E(G)| - |E[G[X]]|$

$|E(G)| = m, \quad |E[G[X]]| = q$

$\mu = m - q \qquad \cdot\, m - q = 0$

So, by induction hypothesis is when let us say that m - q = 0, what is the meaning of m - q = 0? So, here you have an X and here it is Y and there is no edges here, that is what it means that the number of edges is 0. So, look at the total number of edges if I delete the number of edges which are here, they are 0 it means there is no edges between X and Y.

(Refer Slide Time: 24:10)



$\cdot\ \lambda$ fr $(X,Y) = 0$

Uni Important Cut $= \emptyset$

$\frac{1}{4^0} \qquad \frac{1}{2^\lambda} = \frac{1}{2^0}$

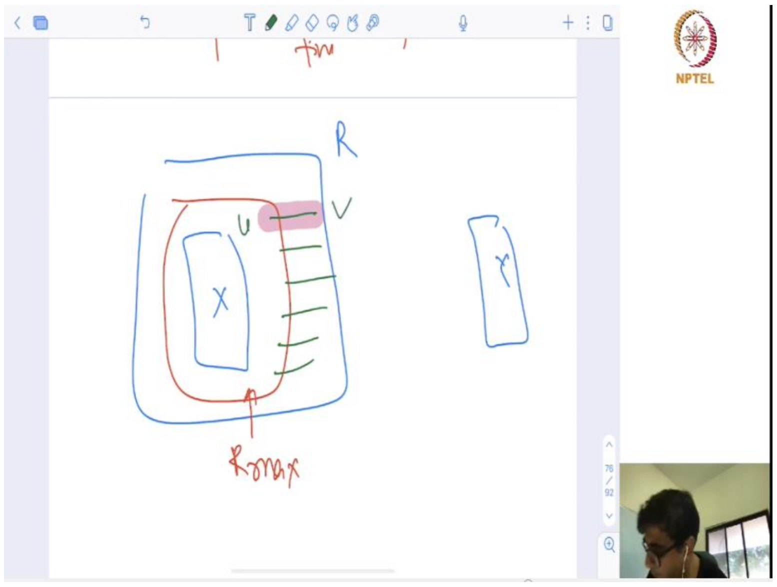
1 fine 1

Lemma

In this case, what is a lambda for X, Y cut is also 0. Then in this case, what is my left-hand side? My left hand so in this case, there is a unique important cut and that has size cut. So, for this I am trying to so this is for base case So, it means what is the left-hand side will be 1 to the power 4 to the power 0. And what is the left-hand side is like 1 by 2 to the power lambda, which is equal to 1 by 2 to the power 0.

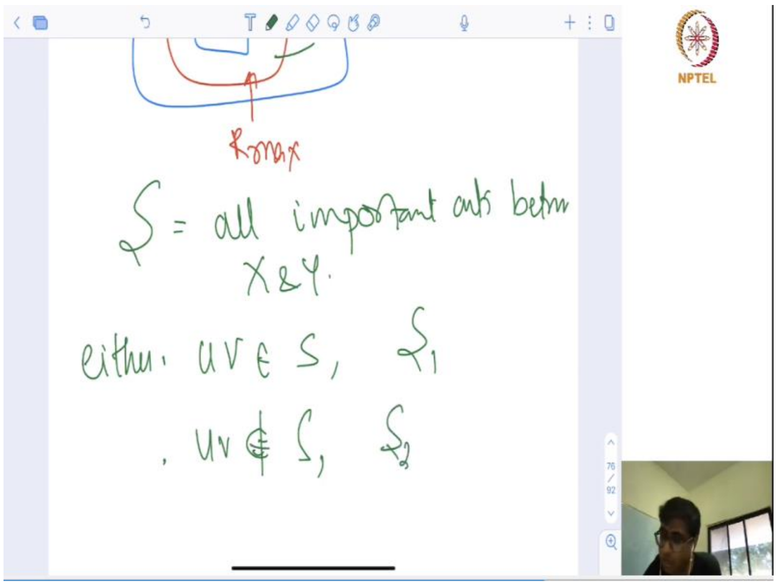
So, both are 1 so this is fine. Let us add an extra phase. So, now you recall, the way every important cut.

(Refer Slide Time: 25:07)



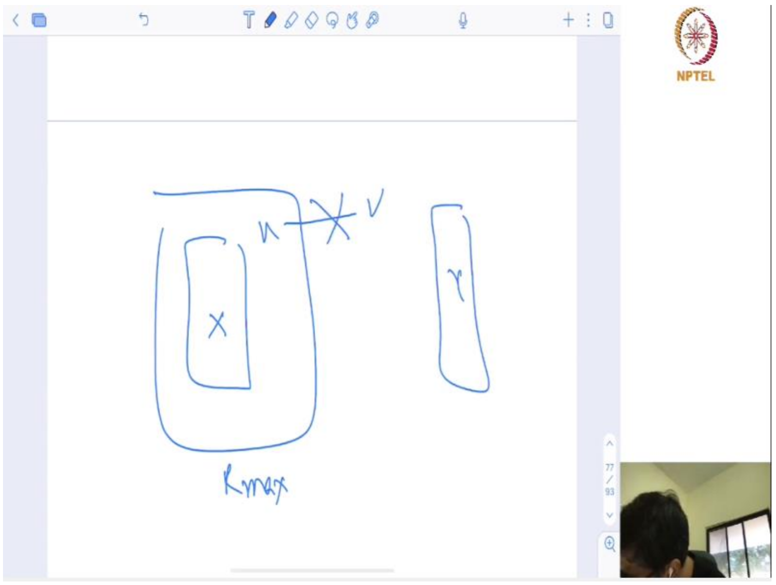
What we knew about any every important cut? Every important cut had the property is the following. So, suppose this is my X and this is my Y, every important cut. Suppose this has R, what are the property? That they contained the following object. This was R max and how did we counted important cuts, we said Hey, look at look at all the edges going across. And let us say this edge and let us call this u and v.

(Refer Slide Time: 25:43)



So, we had two cases so suppose this is my S all important cuts between X and Y. So, either u v belongs to S let us call S 1 or u v does not belong to S u v u v does not belong to my S or u v does not belong to us this is called S 2. So, we will count each of them so now let us see what happens in the first case?

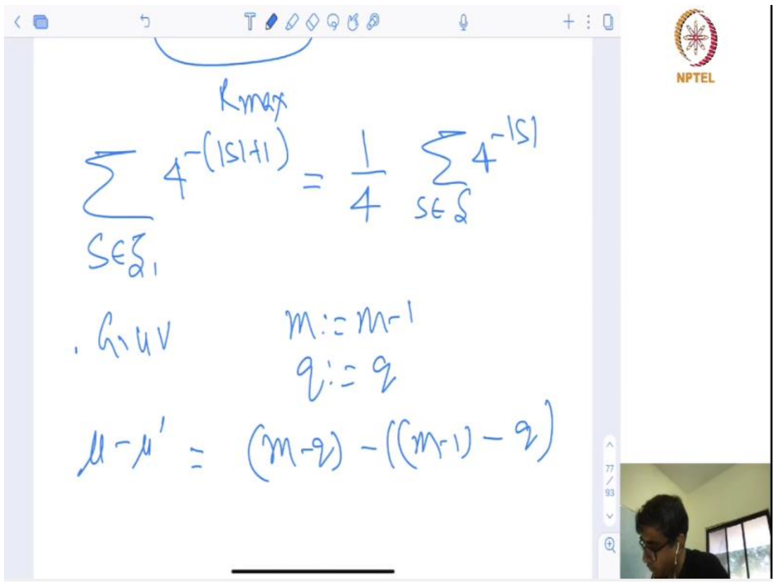
(Refer Slide Time: 26:27)



Well, in the first case when you do not so, here is your graph, this is your X, this is your R max. And here it is your u v you are thrown this your Y. Now notice, we have already shown in our branching that every important cut of G – u v is also an important cut of G like every important cut of G that contains u v is also an important cut of G – u v. So, what are we going to do we are going to give we are going to get all the important cuts of all the important cuts of G – u v and add + 1 for u.

So, now what are we going to get? So, what are we need to bound them? Then we need to bound f belonging to S 1 1 to the power 4 to the power mod S + 1. This is what we have to look. Now notice what happens to R?

(Refer Slide Time: 27:37)



k_{max}

$$\sum_{S \in \mathcal{S}_1} 4^{-(|S|+1)} = \frac{1}{4} \sum_{S \in \mathcal{S}} 4^{-|S|}$$

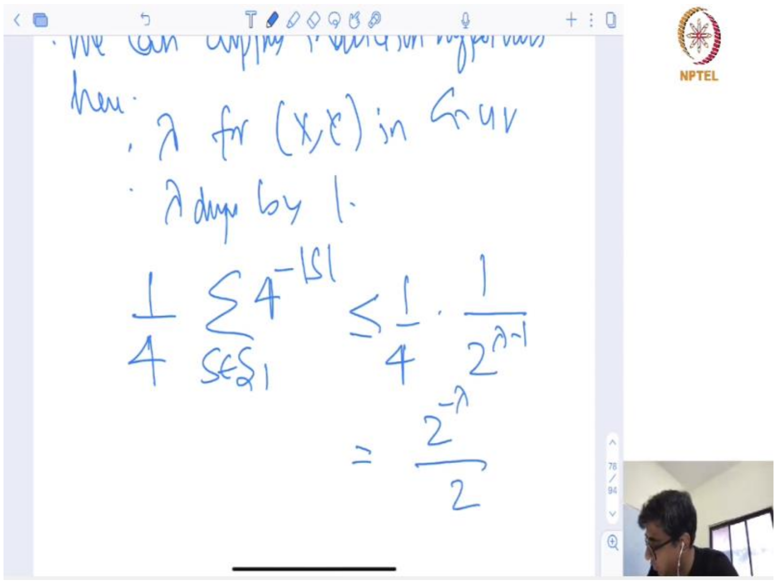
. $G \setminus uv$ $m := m-1$

$q := q$

$$\mu - \mu' = (m-q) - ((m-1) - q)$$

So, this is from we have to upper bound S in S 1 4 to the power – (()) (27:48). Now, which is = 1 by 4 s in S 1 4 to the power – (()) (27:57). Now, notice what happens first of all in G – u v. What is this? m drops by m 1 and the q the edges induced graph and x q remains q. So, the measure mu - mu prime is m - is if m - q - m - u – q.

(Refer Slide Time: 28:53)



. we can apply induction hypothesis here.

. λ for (X,Y) in $G \setminus uv$

. λ drops by 1.

$$\frac{1}{4} \sum_{S \in \mathcal{S}_1} 4^{-|S|} \leq \frac{1}{4} \cdot \frac{1}{2^{\lambda-1}}$$

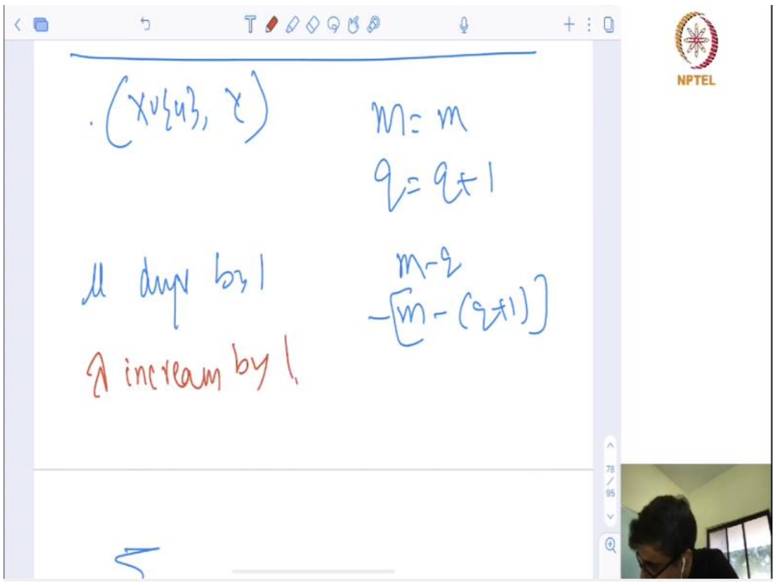
$$= \frac{2^{-\lambda}}{2}$$

And so, this is going to be 1 so major drops by 1. So, we can apply induction hypothesis here. But now, we also have to find what is the lambda for X, Y in G – u v. When lambda drops by exactly 1 it cannot drop by 2 otherwise then you take that add this edge and then you are done. So, by induction hypothesis what this number will be is 1 by 4 summation S in s 1 4 to the power - S which is less than equal to 1 by 4 which is less than or equal to which is less than equal to sorry, 1 by 4 times 0 by 2 to the power lambda – 1.

Which is equal to 2 to the power - lambda divided by 2. So, this is exactly what you are going to get here. This you are able to bound by this which is like 2 to the power - because like this is what we have in this but this is upper bounded by number of all cuts important cuts in G – u v. Which contains this and that is what is bounded by 2 to the power - lambda - 1 by 4 and that is 2 power – lambda.

What you get to look at the other direction other direction what happens? You decide that well u v is not in this.

(Refer Slide Time: 30:43)



$(X \cup \{u\}, Y)$

$m = m$

$q = q + 1$

μ drops by 1

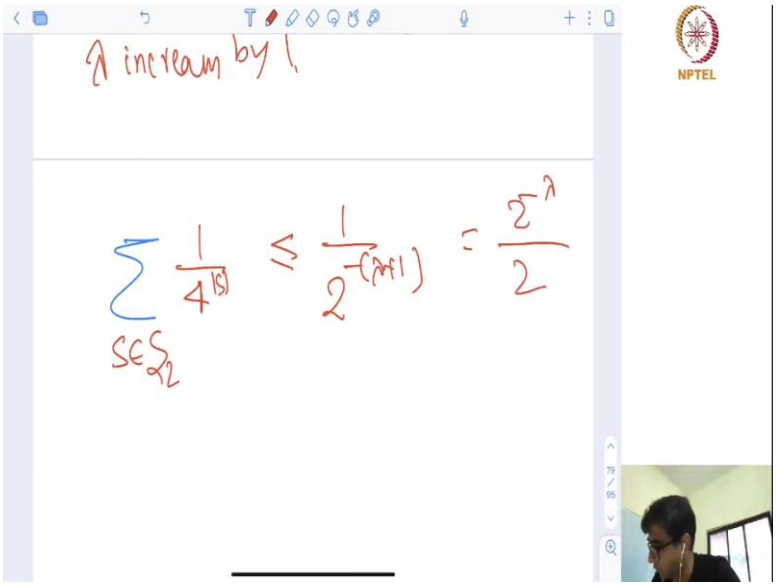
λ increases by 1

$m - q$

$-[m - (q+1)]$

Then what you are looking for is that, in the second case I am looking for all important cuts between this and Y. But now, what happens to this now notice m remains m but number of edges q becomes so mu drops again by 1 because it is m - q - m - k + 1. So, mu drops by 1 and hence what we can say?

(Refer Slide Time: 31:16)

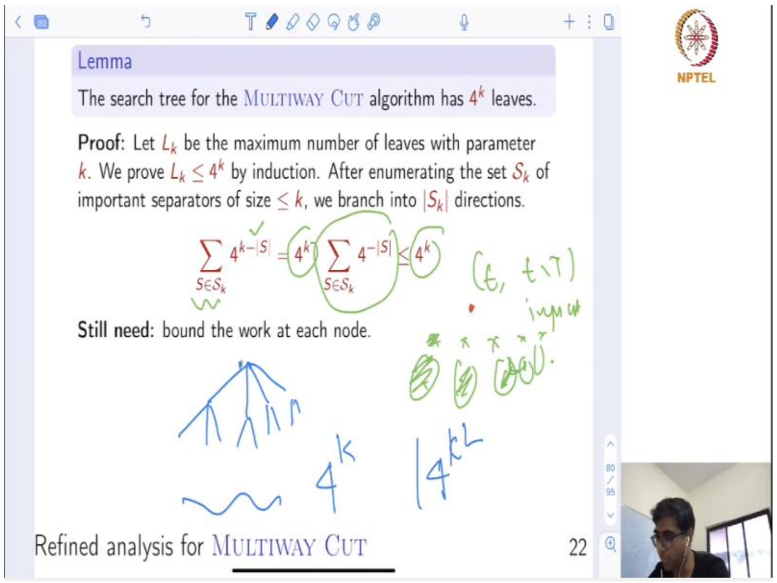

λ incream by 1

$$\sum_{S\in S_2} \frac{1}{4^{|S|}} \le \frac{1}{2^{-(\lambda+1)}} = \frac{2^{-\lambda}}{2}$$

For these things and look in this case what happens in this case because this was an R max the moment you add v to this lambda increases by 1. New drops or 1 but lambda increases by 1 2. So, now for S in S 2 all we can say that 1 by 4 to the power mod s 4 to the power mod s is less than = 1 over 2 to the power - lambda + 1 which is 2 to the power - lambda divided by 2. So, in both sides you are able to bound by 2 to the power - lambda by 2 to the power - lambda 2.

So, you are able to bound the whole thing by 2 to the power - 1. So, this is a refined bound we are able to achieve. Now, how are we going to use this refined bound for our algorithms let us see that.

(Refer Slide Time: 32:38)


Lemma

The search tree for the MULTIWAY CUT algorithm has 4^k leaves.

Proof: Let L_k be the maximum number of leaves with parameter k. We prove $L_k \le 4^k$ by induction. After enumerating the set S_k of important separators of size $\le k$, we branch into $|S_k|$ directions.

$$\sum_{S\in S_k} 4^{k-|S|} = 4^k \sum_{S\in S_k} 4^{-|S|} \le 4^k$$

Still need: bound the work at each node.

Refined analysis for MULTIWAY CUT

22

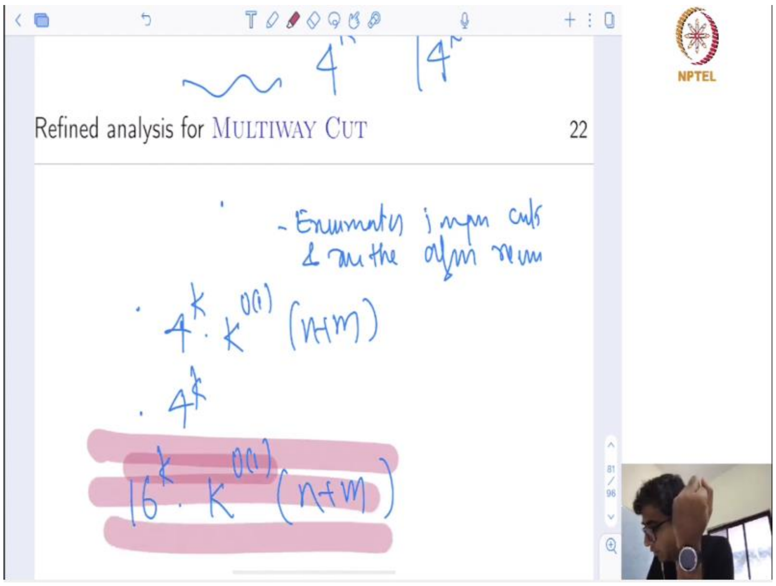
So, now let us look at the search tree for multi way cut algorithm we are going to show that it has 4 to the power k lives. Let L k be the maximum number of leaves with a parameter k. We are going to prove L k by induction. So, how does algorithm works? You have t, T - t you enumerate important cuts and then you that you enumerate important cuts and then you recursively solve the problem.

How many important cuts like so, that is what this algorithm is going to be how many number of leaves going to be. So, parameter drops in each of these branches. So, by induction hypothesis let us see. So, by S in S k the total number of leaves in this is the leaves here + leaves here + leaves here + leaves which is summation S in S k 4 to the power k minus, because that is so many branches for each S in S k the important cuts it drops by k – S.

And so, for this by induction hypothesis number of leaves is upper bounded by 4 to the power k - S the parameter. So, the total sum is 4 to the power k – S S in S k which is 4 to the power k if you take out a S in S k and this you know is upper bounded by 1 so it is 4 2. So, all we are able to do is that not like the we are able to bound the search tree of multiway cut algorithm by four to the power k and not 4 to the power k squared.

Still, we need to bounce we need to bound a lot of work at each node. This does not imply that we have a 4 to the power k algorithm but it immediately implies that we have achieved a better algorithm.

(Refer Slide Time: 34:34)



$\sim\ 4^{k}\quad |4^{k}$

Refined analysis for Multiway Cut

22

- Enumerating impn cuts & run the algm recurs

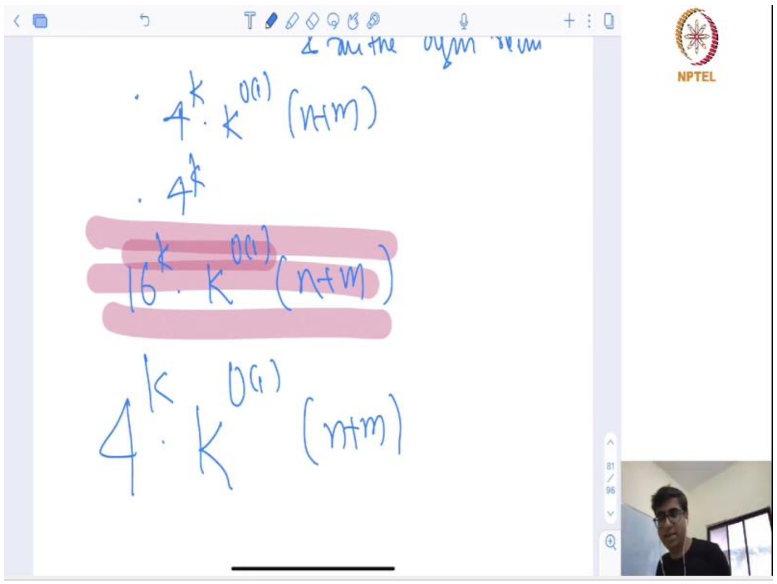
- $4^{k} \cdot k^{O(1)} (n+m)$
- 4^{k}

$16^{k} \cdot k^{O(1)} (n+m)$

Why? Because what is algorithm doing, algorithm enumerate important cuts and run the algorithm recursively. So, at this point at each node at each node the number of the running time of the algorithm can I can only say that it is a 4 to the power k times k to the power big of 1 n + 8. Because that much amount of time to enumerate each important cuts and recursively and since there are a number of nodes is upper bounded by 4 to the power k.

So, the running time of the algorithm will be 4 into 4 16 to the power k k to the power big O of 1 into n + 1. So, this is the running time of the algorithm.

(Refer Slide Time: 35:36)



& run the algm

$\cdot\ 4^k \cdot k^{O(1)} (n+m)$

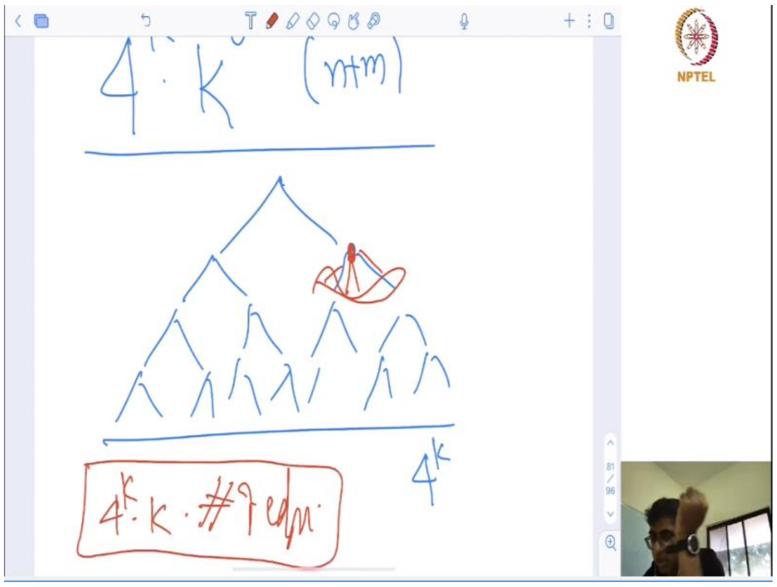
$\cdot\ 4^k$

$16^k \cdot k^{O(1)} (n+m)$

$4^k \cdot k^{O(1)} (n+m)$

But what we promised is an algorithm with running time 4 to the power k k to the power big O of 1 and 2 n +. And to achieve this, it is a bit more complicated, and I will refer you to the book for this.

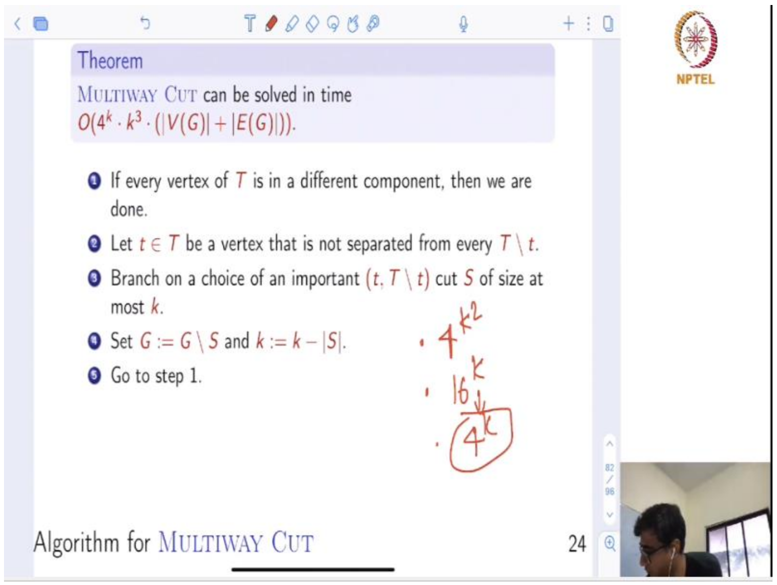
(Refer Slide Time: 35:46)



But let me give you some idea how we achieve these kinds of things. Look at the number of leaves number of leaves is 4 to the power. Now look at work at any node, if we could show that the work is proportional to the number of leaves then we are happy. But the problem is work may not be proportional to the number of leaves. because you do not only enumerate important cuts of what you need for your algorithm.

But you actually enumerate a family of sets and from there you filter out important words. So, there are certainly that we need to handle, but we can handle those activity and leading into the podcast with them. So, ultimately will show that the amount of work everybody does is proportional to the total number of edges in this graph in this tree, which is like 4 to the power k time scheme roughly number of edges. I think that leads us to the following algorithm.

(Refer Slide Time: 37:02)



Theorem

Multiway Cut can be solved in time $O(4^k \cdot k^3 \cdot (|V(G)| + |E(G)|))$.

1. If every vertex of T is in a different component, then we are done.
2. Let $t \in T$ be a vertex that is not separated from every $T \setminus t$.
3. Branch on a choice of an important $(t, T \setminus t)$ cut S of size at most k.
4. Set $G := G \setminus S$ and $k := k - |S|$.
5. Go to step 1.

$\cdot\ 4^{k^2}$

$\cdot\ 16^k$

$\cdot\ \boxed{4^k}$

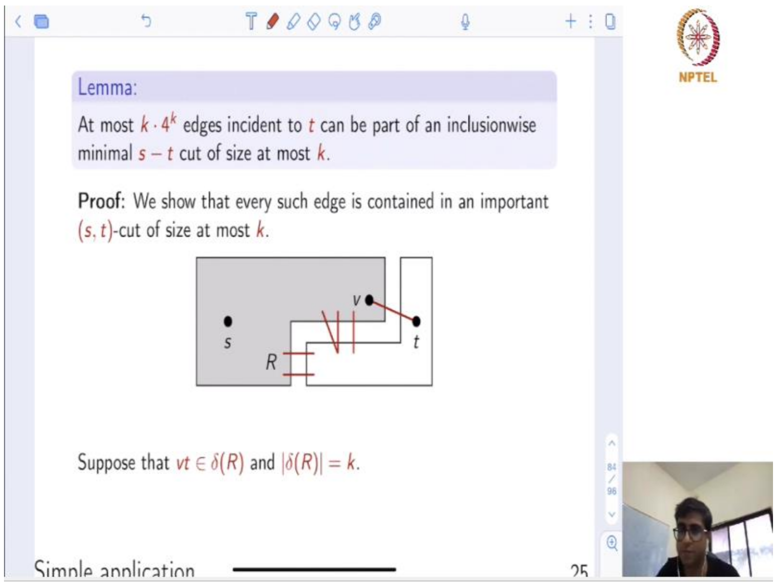
Algorithm for Multiway Cut

24

If every vertex is at a different component, then we are done, let t be a vertex that is not separated from every T, you branch on the choice of report and cut, you really do and say decrease the parameter go to step one. So, this is the algorithm we had in the beginning also, but we could only, so we first k 1 algorithm 4 to the power k square dependence on k then 16th of R k.

And we I gave you just very briefly I will say nothing like that you can get from 16 to the power k 4 power k with more refined and measured analysis of the whole algorithms, but I leave that for you to make work out.

(Refer Slide Time: 37:46)



Lemma:

At most $k \cdot 4^k$ edges incident to t can be part of an inclusionwise minimal $s - t$ cut of size at most k.

Proof: We show that every such edge is contained in an important (s, t)-cut of size at most k.

Suppose that $vt \in \delta(R)$ and $|\delta(R)| = k$.

Simple application

25

And there are some other simple applications of important cut which I will not talk but like you can look into these slides and we will just leave it there because they are very useful and very good. Like for example, you can show that you give you fixed annuity and look at the degree look at the number of edges which are incident to about s, t. Consider how many edges I like are a part of minimum s-t cut of size at most k.

At mostly incident to T can show that is also some function of k in before the R k again using an application of important cuts, but I will leave that there. And with that, we will end this lecture and we will try to see another application of important cuts in directed feedback What is it and why it cannot be applied to other cases of multiway cut like other generalization of multiway such as multi cut in the next couple of lectures of the (()) (38:38) lectures. So, thank you.