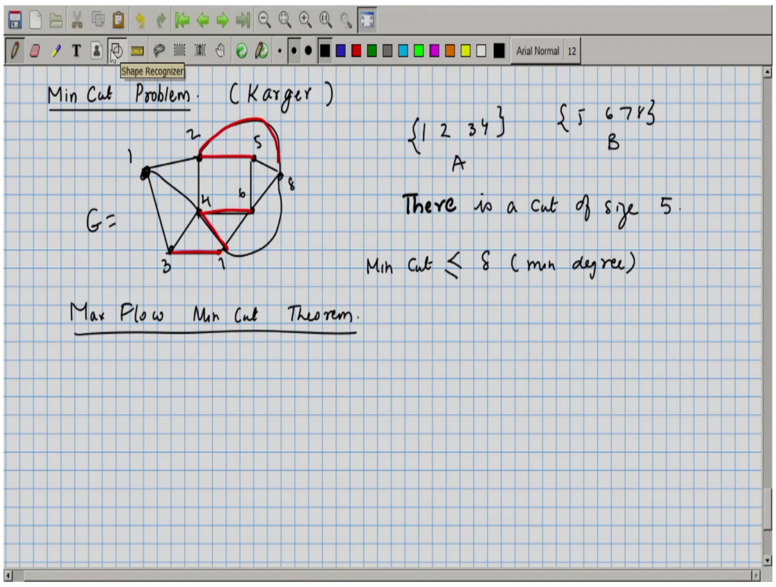
# Randomized Algorithms

**Prof. Benny George Kenkireth**
**Department of Computer Science & Engineering**
**Indian Institute of Technology, Guwahati**

## Lecture – 02
## Randomized Mincut Algorithm

(Refer Slide Time: 00:28)



Min Cut Problem. (Karger)

$G =$

$\{1\ 2\ 3\ 4\}$ $\{5\ 6\ 7\ 8\}$
A B

There is a cut of size 5.

Min Cut $\leq 8$ (min degree)

Max Flow Min Cut Theorem.

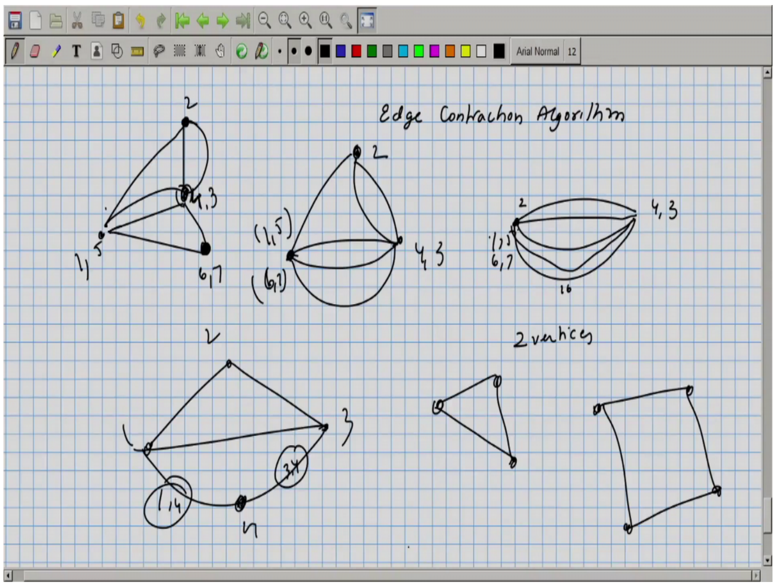
We will see an algorithm for a particular problem. So, this is the Mincut problem and the solution that we will see is an elegant solution given by Karger. So, what does the min cut problem? Let us understand what a cut is. So, we are given a graph and we need to find a collection of edges such that if you remove those edges the graph becomes disconnected ok. So, in this graph what is the collection of vertices edges that you could remove to make this graph disconnected. We can also think of it in the following way, just split the collection of vertices into 2 parts. So, here 1 to 8 of the vertices we could take some vertices on one side, let us say if you take 1 2 3 and 4 on one side and 5 6 7 eight on the other side.

Look at the edges that are going from this set to this set those edges will be called the cut edges. So, here let us just mark out the cut edges in a different color ok. So, all the edges that are in black are either from A to A or from B to B. So, here there are 1 2 3 4 5 edges. So, we can write that observation as there is a cut of size 5. So, this is a cut with exactly 5 edges in it. So, we can ask this question amongst all possible cuts in the graph which is a cut of smallest size, there is a complimentary problem which is what is the cut of the maximum size.

Question that we will address today is the min cut problem and we will come up with a randomized algorithm to find the min cut. So, in this particular graph what is the min cut and one thing we can say is the min cut size should certainly be equal to the minimum degree of the graph. So, look at all vertices this is a vertex of degree 3, this is a vertex of sorry the minimum cut need be no greater than smaller than delta smaller than or equal to delta, the delta is the min degree ok. Because, if you remove those many you can certainly disconnect that particular graph at that particular vertex. So, here there will exist a cut of size 3, we can remove 3 edges mainly 1 3 1 4 and 1 2 and get a suitable cut.

But are there smaller cuts, is there a cut of size 2 and we can verify that there will not be any cut of size 2 in this graph because every edge is part of some cycle ok. Therefore, since every edges part of some cycle removing that edge will still preserve at least one part in the graph ok. So, in this graph the minimum cut will be of size 3. There is a theorem called max flow min cut theorem, but we will find the solution to this problem, an algorithm to this problem without having to go through max flow min cut theorem. So, this theorem says that the flow maximum achievable flow in a capacitated network is equal to the size of its minimum cut.

(Refer Slide Time: 05:50)



So, let us see how this algorithm works. We will first see it with an example. So, consider this graph on 7 vertices. What we will do is we will call our algorithm as the edge contraction algorithm ok. So, we will choose an edge let us say 4 3 and then contract that edge. What is this contraction of operation? So, here we will take this vertex 3 and 4 the edge between 3 and 4 and then put this over here ok. So, this edge goes at 4 there will be vertex 3 2 and 3 were connected they will remain connected 4 and 7 will now be connected. So, that you will get a super node 4 3 at that vertex.

Now, you could choose another vertex let us say 5 and look at the edge 5 1. So, we will remove this and contract this particular edge when we do that we will essentially get one particular; so, this will come all the way till here. So, 5 1 5 will be this and this edge would be there and this edge would also be there and then we will choose another. So, we will continue this process repeatedly ok. So, if we contract 6 and 7 then we will get 6 7 here and we will be left with this particular edge. Now, if we contract let us say this vertex I mean this particular edge we are always contacting an edge; if you contact this edge then we will get the following graph 2 4 3. There is this 2 edges here, 1 5 was 2 edges already present and this will be 6 7 being added to this, there is going to be an edge of this kind.

Now, if you contract these two we will get 2 4 3 1 5 6 7 these 3 edges would be present ok. So, after doing this process we have ended up we ended up with some graph consisting of 2 vertices. Look at the edges across them each edge we could have named each of these contracted edges they had a certain name. For example, if we had named the edges initially this is the say 1 6 edge we could just put this as 1 6 edge. So, after all the contraction we are we do not have to worry about which edge did it originally correspond to. So, when we reach a configuration with 2 vertices look at the number of edges between these 2 vertices that would basically be a cut. Now, the question is will this be the minimum cut?

What is the probability that this is the minimum cut? ok. So, I hope the algorithm was clear to all of you. What we did is the following, we had a graph it had let us say m edges to start with choose any particular edge uniformly at random. So, each edge has a probability of 1 by m being chosen as the first edge to contract. Once that choice is being made the end points of those 2 vertices of those 2 of that edge is being joined together or we can think of that vertex forming on the edge being contracted to get a super node ok.

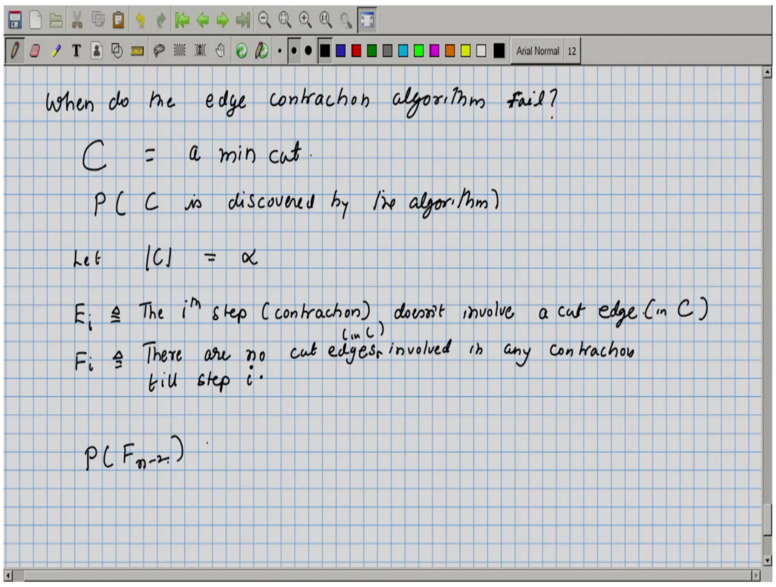
So, that is a that is the contraction step. This is being done repeatedly till we are in a situation where, we have precisely 2 vertices. At that point we look at the all the edges, we could for each edge at the start itself associate the name of the edge. For example, in let us say if this is our graph we could call this edges a 1 4 edge and label it and this as the 3 4 edge.

So, when the contraction is being done we do not lose information corresponding to which was this edge in the original graph. So, once the contraction process is finished and we end up with exactly 2 vertices we can from the collection of edges figure out which where these edges in the original graph. We have to answer the following question: first question being will this collection of edges be a cut. Of course, it will be a cut because, it is certainly going to separate out the 2 vertices or the 2 super vertices they will correspond to certain vertex set in the original graph. So, these edges will be the cut edges for those partition of the vertices. Second more important question is will it be the will it be a minimum cut. There could be more than one minimum cut.

For example, if you take a cycle if you take a 3 cycle any of the edge I mean any of the 2 edges is going to be a cut ok. They are the minimum cut ok. In this graph if you take any 2 edges that is going to form a cut and that is the minimum cut and there are many minimum cuts ok. So, is there a non-zero probability of finding the minimum cut? The edge contraction algorithm may succeed with some probability, what is that probability? If it is an extremely small probability then we do not really gain anything, but if it is a reasonable profit probability; what is extremely small and what is reasonable we will specify it in a while.

But, it if it is a reasonable probability so, let us say something like 1 by a polynomial then we can repeat this algorithm multiple times. And then hope that repeating the algorithm will I mean, if the if the repetitions are independent then the repetitions will bring down the error probability to whatever quantity we want and that is what really happens. So now, what is remaining is we need to analyze this particular algorithm and show that it has a significant probability of success.

(Refer Slide Time: 14:47)



When do the edge contraction algorithm fail?

$C$ = a min cut

$P(C$ is discovered by the algorithm$)$

Let $|C| = \alpha$

$E_i \triangleq$ The $i^{th}$ step (contraction) doesn't involve a cut edge (in $C$)

$F_i \triangleq$ There are no cut edges (in C) involved in any contraction till step $i$.

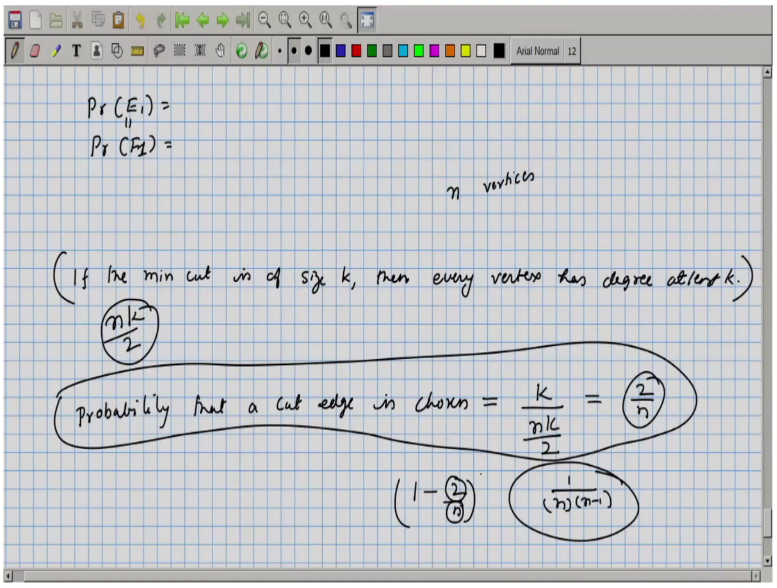
$P(F_{n-2})$

When does the edge contraction algorithm fail? If we choose a cut edge at any stage during our algorithm then the algorithm is not going to find a minimum cut ok. So in fact, we will look at the instead of computing the probability of the algorithm failing we will fix a particular cut C and we will compute the probability that the algorithm finds this particular minimum cut. So, if C is a minimum cut we will compute the probability that our algorithm finds this particular minimum cut. And that probability is going to be surely a lower bound on the probability of success. Because, even if the algorithm had found another minimum cut different from C the algorithm would have succeeded ok. So, if we say that this probability itself is significant; that means, overall probability of success is significant. So, we want to find the probability that C is discovered by the algorithm.

So, let us look at certain events, we will split this particular big event that C is discovered by the algorithm, it has many many small events. Let size of C be equal to alpha ok. So, since it is a minimum cut we know that alpha is small or we can expect that alpha is small. So, when the algorithm is choosing an edge from I mean at random from the complete collection of edges, the probability that it will pick an edge from this particular collection will be small. We will show that cumulatively the error that accumulates in various steps of the algorithm is not so, significant to make the algorithm perform really poorly. So, let us say E i is the even that the ith step or the ith contraction does not involve a cut edge ok.

And, F i denotes the following event the ith step there are no cut edges involved in any contraction till step i ok. So, here when I say does not involve a cut edge I mean a cut edge in C; the same here no cut edge in C. What I am really interested in is the probability of F n minus 2. So, we need to compute the probability of F n minus 2. So, how do we compute this? So, let us look at the probability that in the very first step we will choose.

(Refer Slide Time: 19:08)



$Pr(E_1) =$

$\parallel$

$Pr(F_1) =$

$n$ vertices

(If the min cut is of size $k$, then every vertex has degree at least $k$.)

$\frac{nk}{2}$

Probability that a cut edge is chosen $= \frac{k}{\frac{nk}{2}} = \frac{2}{n}$

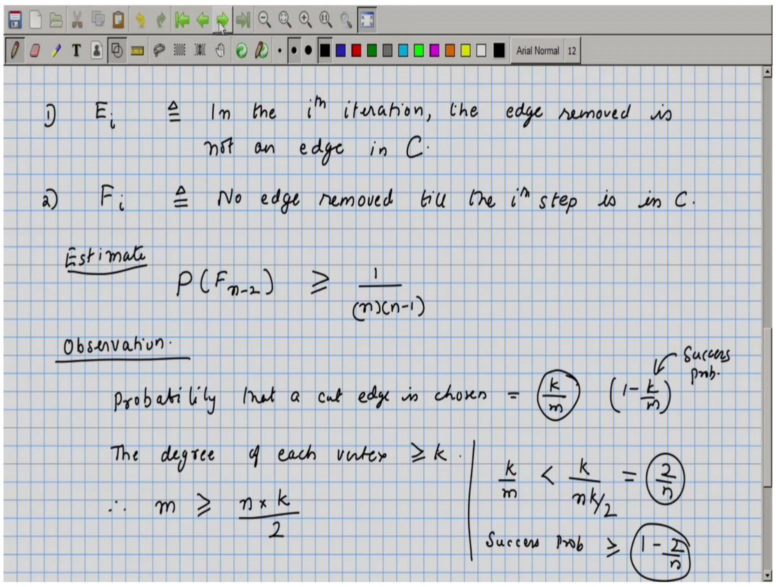
$\left(1 - \frac{2}{n}\right)$ $\frac{1}{(n)(n-1)}$

So, probability of E 1 ok, what is this we know that probability of E 1 and probability of F 1 is the same ok, this is these are the same ok. So, how do we compute this? Since, our graph had let us say n vertices and if the size of the cut is k, if the min cut is of size k then every vertex has degree at least k. If this was not the case then we could get a smaller cut. So, the min cut is of size k every vertex has degree at least k; that means, the total number of edges is at least n k by 2 each edges counted for 2 vertices.

So, this is the total number of this is a there are at least these many edges in the graph. So, the probability that one of these k edges is being chosen is. So, probability that a cut edge is chosen is equal to k by n k by 2. So, this gives us 2 by n as the probability that a cut edges chosen. The probability that no cut edge, what we want is probability that a cut edges not chosen that is going to be 1 minus 2 by n ok. Now, as the algorithm progresses these values are going to be changing, but we will see that the idea that we used here can essentially be extended using the method of conditional probabilities and we can compute that. And in the end we will get we can bound the probability by something like 1 in 1 by n into n minus 1.

So, we can show that we will have at least this much probability of success by an argument very similar to the argument that we did here. So, at each intermediary step if so far there was no cut edge involved in any of the contraction then every remaining super node will have degree at least equal to the cut size and that helps us compute the probability that a cut edge is chosen; when you choose randomly from the collection of all remaining edges. What we are interested in is a probability that we never contract an edge corresponding to a cut C. Let us say C was a fixed minimum cut, we will compute the probability that the iterative steps never remove an edge from C.

(Refer Slide Time: 22:46)



1) $E_i \triangleq$ In the $i^{th}$ iteration, the edge removed is not an edge in $C$.

2) $F_i \triangleq$ No edge removed till the $i^{th}$ step is in $C$.

Estimate

$$P(F_{n-2}) \geq \frac{1}{(n)(n-1)}$$

Observation.

Probability that a cut edge is chosen $= \boxed{\frac{k}{m}}$ $\left(1-\frac{k}{m}\right)$ Success prob.

The degree of each vertex $\geq k$.

$\therefore\ m \geq \frac{n \times k}{2}$

$\frac{k}{m} < \frac{k}{nk/2} = \boxed{\frac{2}{n}}$

Success Prob $\geq \boxed{1-\frac{2}{n}}$

We had introduced certain events, the first event that we described was E i. So, E i is this event that in the ith iteration there is a particular edge being removed that edge is not a cut edge. The edge removed is not an edge in C. The second event that we described was F i which denotes the intersection of all E i's up till that stage, in other words till the ith stage none of the edges removed belongs to the cut. So, this is the event that no edge removed till the ith step and including the ith step is in C.

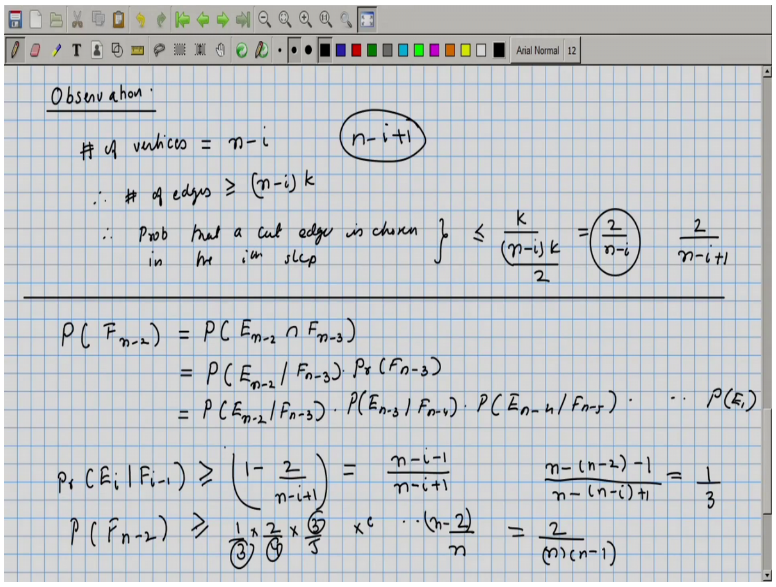
What we are interested in estimating is the following, probability of F n minus 2 that is we had contracted n minus 2 edges when we contract an edge we have one less vertex. So, in the end we have when we have performed n minus 2 operations, if we start with the graph with n vertices we would end up in a situation where we have precisely 2 vertices. These vertices are super nodes and therefore, those super nodes define a certain cut and we are interested in the probability that that cut there is nothing, but C. So, this is the probability that we are interested in and we will show that this probability is greater than 1 over n into n minus 1. The intuition behind this is since, we are looking at the smallest cut the probability that you will remove an edge belonging to the smallest cut, we will argue that in each of these iterative steps that probability is small.

So, together their probability is not significantly high in order to make the algorithm perform poorly ok. So, how do we compute this probability? So, let us look at these cut edges. So, at the very start the probability that if you choose a random edge the probability that it is a cut edges going to be probability that a cut edge is chosen is k divided by m. So, we have 1 minus k over m as the probability of success for the first step ok. We want to argue something about this k by m, here the dependency is between the number of cut edges and the number of edges in the graph. We will relate it to the number of vertices in the graph. How do we do this?

So, since we know that the smallest cut we are assuming that the smallest cut contains k edges every other cut must surely have more than k edges greater than or equal to k edges. So, we can make the following observation. The degree of each vertex is greater than or equal to k because, if there was even one vertex whose degree is less than k you can choose that vertex, remove these edges and get a cut of size smaller than k. So, that is not possible therefore, degree of each vertex is greater than or equal to k. Therefore, we can argue that the number of edges should be greater than total number of vertex into the minimum degree of each vertex by 2, by 2 because each edges counted for every for 2 vertices ok.

So, m is greater than n k by 2 therefore, we can conclude that k by m is going to be less than k divided by n k by 2 ok. So, this is equal to 2 by n. So, probability that a cut edge is chosen at the very first step is only 2 by n and therefore, the probability that probability of success is; so, success probability is for the first step is going to be greater than 1 minus 2 by n. Note that a similar statement holds true during every iteration.

(Refer Slide Time: 28:32)



Observation:

$\#$ of vertices $= n-i$ $\qquad \boxed{n-i+1}$

$\therefore$ $\#$ of edges $\geq (n-i)k$

$\therefore$ Prob that a cut edge is chosen in the $i^{th}$ step $\Big\} \leq \dfrac{k}{\frac{(n-i)k}{2}} = \boxed{\dfrac{2}{n-i}} \qquad \dfrac{2}{n-i+1}$

$$P(F_{n-2}) = P(E_{n-2} \cap F_{n-3})$$
$$= P(E_{n-2} / F_{n-3}) \cdot P_r(F_{n-3})$$
$$= P(E_{n-2}/F_{n-3}) \cdot P(E_{n-3}/F_{n-4}) \cdot P(E_{n-4}/F_{n-5}) \cdot \cdots P(E_1)$$

$$P_r(E_i | F_{i-1}) \geq \left(1 - \frac{2}{n-i+1}\right) = \frac{n-i-1}{n-i+1} \qquad \frac{n-(n-2)-1}{n-(n-i)+1} = \frac{1}{3}$$

$$P(F_{n-2}) \geq \frac{1}{\cancel{3}} \times \frac{2}{\cancel{4}} \times \frac{\cancel{3}}{5} \times^{c} \cdots \frac{(n-2)}{n} = \frac{2}{(n)(n-1)}$$

So, let us look at that. So, let us say after i iterations we have removed i edges, we are at a stage where we have. So, number of vertices is equal to n minus i and each of these vertices should have degree at least k ok. Therefore, number of edges in the remaining graph is going to be greater than n minus i times k. Therefore, probability if you sample uniformly from all the edges; probability that a cut edge is chosen in the ith step this probability is going to be less than k divided by total number of edges which we know is at least n minus i into k by 2. So, this is equal to 2 by n minus i ok.

So now, let us formally do the analysis of our algorithm. What we are interested in is the probability that probability of the event F n minus 2; till the n minus second iteration no cut edge has been chosen. So, this is equal to probability that no cut edge has been chosen till the n minus third iteration and in the n minus second iteration the edge that is chosen is not a cut edge. So, this we can write it as the probability of E n minus 2 intersection F n minus 3. F n minus 3 denotes the event that no edge chosen till the n minus third iteration has been a cut edge and E n minus 2 denotes the condition or the event that no cut edge is chosen during the n minus second iteration. This we can write it as probability of E n minus 2 given F n minus 3 into probability of F n minus 3.

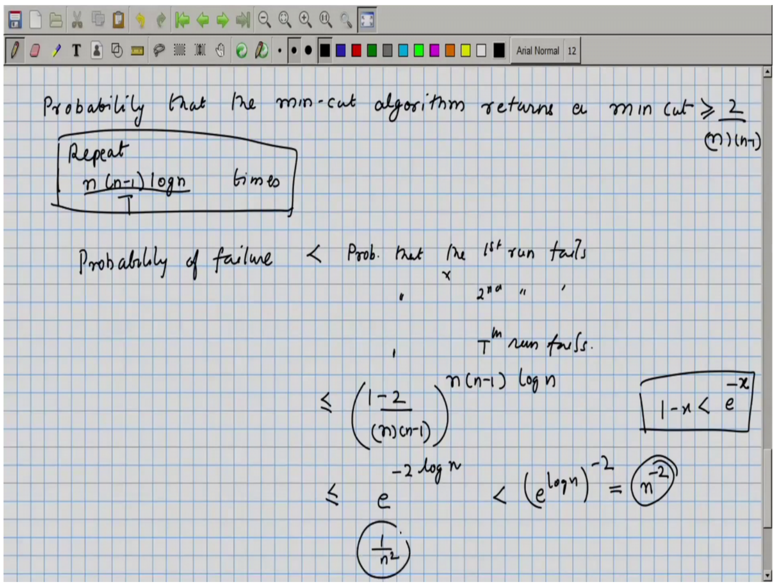
Now, if we know that till the n minus second stage till the n minus third stage no cut edge was chosen, what is the probability that you choose a cut edge in the n minus second stage that is only 2 divided by n minus n minus 3 ok. So, we can expand this, this is equal to probability of E n minus 2 given F n minus 3 into probability of. So, F n minus 2 is this conditional probability times F n minus 3, we can again write this as probability of E n minus 3 given F n minus 4 into F n minus 4 which we can write it as probability of E n minus 4 given F n minus 5. And all the way up to probability of E 1 and E 1 and F 1 they are the same events and their probability we know is going to be bounded by this quantity 1 minus 2 by n ok.

So, we can write this probability is surely greater than this is a probability of success. So, this would be 1 minus 2 by n ok. So, we can write it as n minus 2 by n into n minus. So, what does probability of E i given F i minus 1. So, this is the probability that the edge chosen as a proper edge is not a that it is not a cut edge given that no cut edge belonging to C has been chosen so far. So, this probability is surely greater than 1 minus 2 by n minus i. So, this is going to be equal to n minus i minus 2 divided by n minus i. So, till the i minus first stage we have removed i minus 1 edges, the number of vertices that remain is n minus i plus 1. So, the probability that a cut edge is chosen is only so in the; so, there is a small correction that we will have to do in this particular analysis.

The number of vertices where, n minus i then the probability at any stage by layer sampling; if the probability that a cut edge is chosen is going to be 2 by n minus i ok. So, when we are in the ith iteration of our algorithm the number of vertices that would remain at that stage is going to be n minus i plus 1. For example, at the very first iteration we have n vertices ok. So therefore, the probability will be probability of failure will be 2 divided by n minus i plus 1 ok. So, the probability of success would be greater than; so, probability of E i given F i minus 1 will be greater than 1 minus 2 divided by n minus i plus 1.

This is equal to n minus i minus 1 divided by n minus i plus 1 ok. So, probability of F n minus 2 will be greater than or equal to the first quantity the value of i is n minus 2 there. So, it will be n minus n minus 2 minus 1 divided by n minus n minus 2 plus 1 that is going to be equal to 1 by 3. So, the entire product is going to be greater than 1 by 3 into 2 by 4 into 3 by 5 all the way up to 2 by n minus 2 by n. So, this will be greater than or this will be equal to 2 divided by. So, these terms can 3 and 3 cancels 4 and 4 cancels and so on. So, what will remain is 2 divided by n into n minus 1. So, we can conclude that this algorithm will succeed with this much probability.

(Refer Slide Time: 37:01)



Probability that the min-cut algorithm returns a min cut $\geq \frac{2}{(n)(n-1)}$

$$\boxed{\text{Repeat } \underbrace{n(n-1)\log n}_{T} \text{ times}}$$

Probability of failure < Prob. that the 1st run fails $\times$ 2nd " , ... $T^{th}$ run fails.

$$\leq \left(1-\frac{2}{(n)(n-1)}\right)^{n(n-1)\log n}$$

$$\boxed{1-x < e^{-x}}$$

$$\leq e^{-2\log n} < \left(e^{\log n}\right)^{-2} = n^{-2}$$

$$\frac{1}{n^2}$$

So, probability that the min cut algorithm returns a minimum cut is going to be greater than 2 divided by n into n minus 1. Now, here is a trick that we will do always in our study of randomized algorithm: when we get a significant probability of success, here we call it significant because the denominator is only a polynomial. If the denominator was large, if it was exponentially large then we will not call the probability of success as significant. Here, we get the probability of success being significant. So, what we will do in such cases is we will repeat the algorithm ok. How many times we have to repeat this algorithm to get a reasonably good success probability?

So, let us say we repeat this algorithm n into n minus 1 times log n time. So, this is the number of repetitions ok. So, repeat these many times and amongst all these repetitions find the one which is maximum and which is the smallest number of find the cut which is the smallest number of edges. So, the probability; so, that is our final algorithm and this final algorithm its probability of success is going to be bounded by I mean, it is going to be at least 1 minus failure probability raised to T, where T is the number of times we iterate. If our algorithm had a failure chance of failure sorry ok. So, we are going to repeat our algorithm these many times.

Now when does our algorithm fail? Our algorithm will fail only, if every run of the algorithm fails; even if one of the runs the algorithm succeeds we will say that our algorithm has succeeded. How do we know whether our algorithm has succeeded? Well, what we did is we are going to declare the best possible cut that we have found in these iterations as our cut ok. Now, this algorithm will fail if and only if the best possible cut found in every step was different from C ok. So, that happens with a certain probability. So, the probability of failure is going to be, there is going to be less than probability that the first run fails into probability that the second run fails into probability that the Tth run fails ok. So, this is going to be less than or equal to 1.

So, minus this is the success probability 1 minus that would be the failure probability 2 divided by the whole raise to n into n minus 1 times log n ok. So, 1 minus x is surely less than e power minus x that we apply this approximation, we will get this quantity to be less than e power minus 2 log n ok. So, that is going to be less than e raised to log n the whole raise to minus 2 which is equal to n raised to minus 2 ok. So, the now the so earlier we had the success probability to be greater than something. Now, by repeating the algorithm we have shown that the failure probability is less than 1 by n square.

So, that concludes our analysis of the minimum cut algorithm.