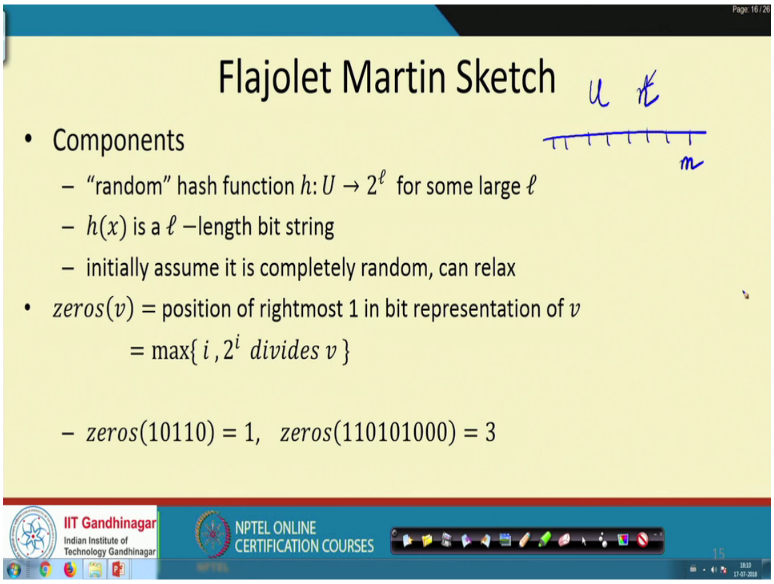
Scalable Data Science
Prof. Anirban Dasgupta
Department of Computer Science and Engineering
Indian Institute of Technology, Gandhinagar

Lecture -08 b
Flajolet Martin + kMV

Hello everyone and welcome to our course on Scalable Data Science. This is a lecture on estimating distinct values. And we look at two algorithms here; the one by Flajolet-Martin and another one on that that uses k Minimum Values. My name is Anirban and am a faculty at Computer Science and Engineering at IIT, Gandhinagar.

(Refer Slide Time: 00:40).

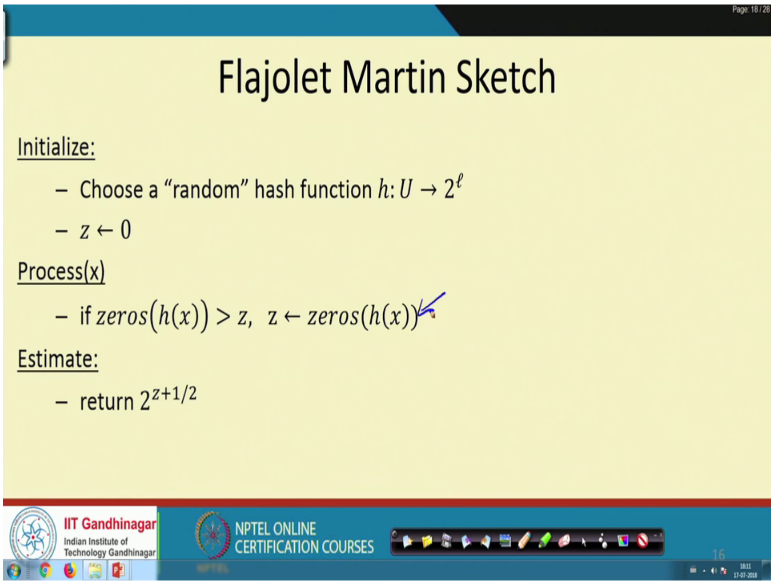


So, we did look at the we did look at the question of defining the question of a estimating distinct values that we have elements from a universe u. And we get n elements from the universe u and given a stream of length m in which possibly duplicates are there right. The length of the stream is m. We want to estimate the number of distinct elements n that is appearing ok.

And we want to do it in space that is much much less than n or the carnality of u. That it ideally should only depend poly logarithmically or logarithmically on these numbers. And we last in last class, we looked at this amazing algorithm by Flajolet-Martin in which or we are we start with the function h that is mapping all elements U into 2 l bit streams with interact the position of the right; most one in the bit representation of each of this hash values right. And for instance here, this position the position in question is this and here the position in question is this right. So, this is 1 and this is 3, because we are using a zero indexing numbering of the positions ok.

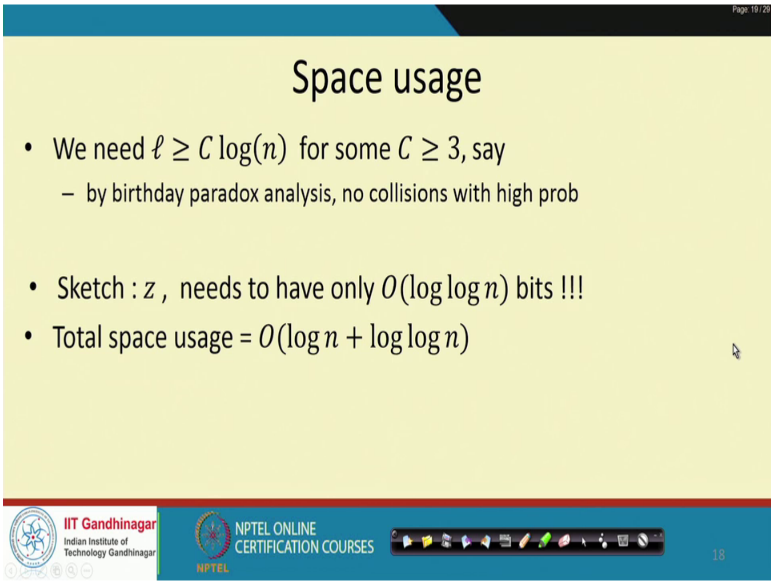
So, let us see. So, now, we did looked at some amount of intuition for this; however, let us also let us not try to do formal analysis of this question.

(Refer Slide Time: 02:04)

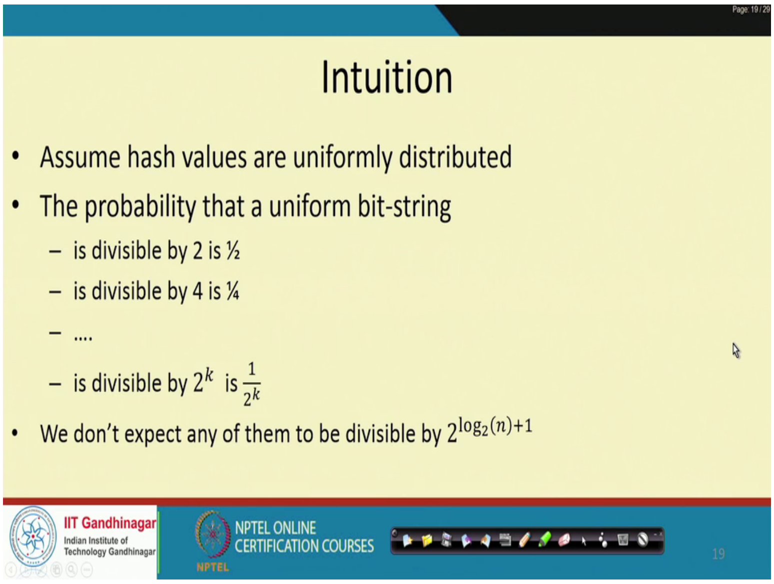


And this is the this is the exact algorithm in case of hand you forgot, me it is very simple the only important line is the only important line is this particular update of z; the z tracts the position of the of the of the maximum of the rightmost one the maximum of these positions ok. And we and then we return it, the estimate that is based on z ok.

(Refer Slide Time: 02:38).

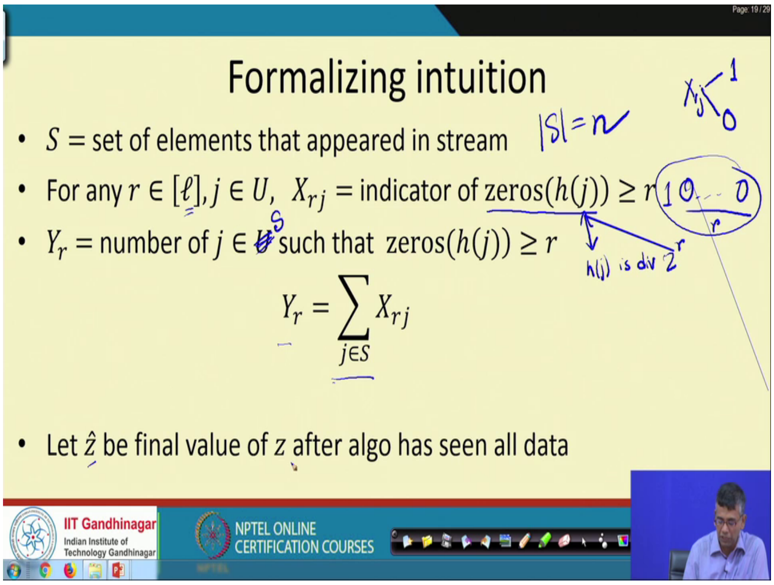


(Refer Slide Time: 02:40)



We track the space usage of this and sum intuition that let us try to formulize it.

(Refer Slide Time: 02:41)

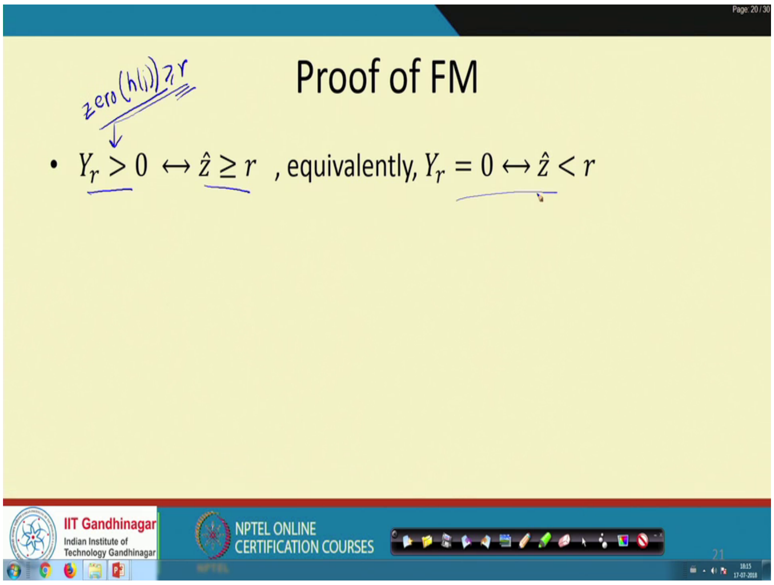


So, suppose S is the set of elements that have appeared in this stream right. So, so according to annotation we just kneeling down to notation carnality of S equals n. Now for any so, remember l is the is the size of the bits stream that the that is the range of the hash function. So, for any r in l and for any element j gets belongs to U, let X r j be a indicator random variable so; that means, that X r j is 1 right. If the zeroes of h of j is at least r which means that h of j has at least which means that h of j has at least r zeroes here right. And then it may be some more zero and then a one ok.

If right in that case you set that X r j to be X r j to be 1. If is this happens, else you set X r j to be 0. So, X r j is this is this indicator variable and let Y be the count of how many of the j's have zeroes of h j to be bigger than r, which means that remember that this also means that this I mean this notation also means that h of j is divisible by is divisible by 2 to the r right ok.

So, so, now let Y r be the count of that right. So, so it is very simple to see that Y r is nothing, but the summation of j belongs to sorry this should actually be S right. This is in that in the stream of course. So, it is not really hard to see that Y r is really the summation of j belongs to S X r j because, this contributes either a 0 or 1 for a for every elements j and Y r is exactly that count. And just last bit of notation that z hat to be the value of z after the algorithm have seen all the data. So, the so, the algorithm is maintaining is maintaining this z hat right is maintaining the z hat right and z hat and this counter z and z hat is the final value of z that is returned here. Let us sort of make some very simple statements.

(Refer Slide Time: 05:20)

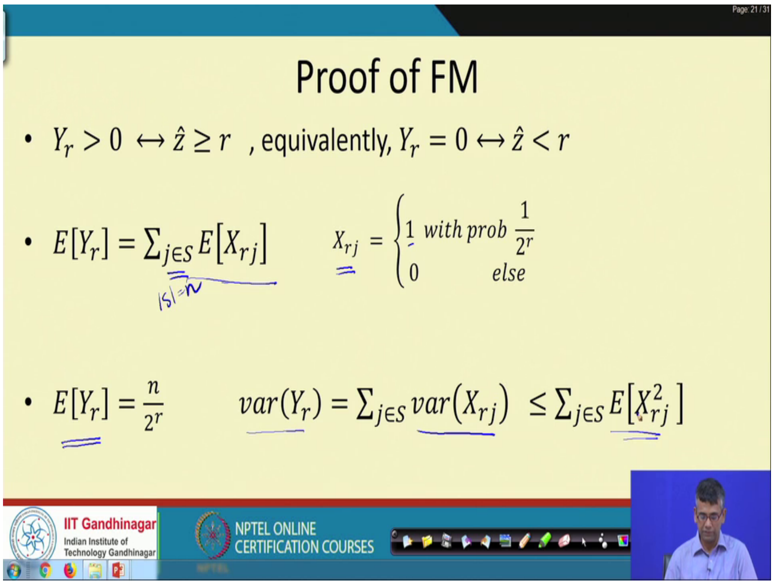


So, first we say that if Y r is strictly bigger than 0, this happens if and only if z hat is at least r. So, let us parse these two states parse of the statement. So, Y r strictly bigger than 0 means what? Means that there is at least one element j that has appeared in the stream j belongs to Sl there is at least one j belongs to S right that has 0s of h of j to be greater than or equal to r right. So, the which means the I mean let me write it down, there exists at least one j that has 0s of h of j to be greater than equal to r. And if that happens, then z hat automatically is also greater than equal to r because z hat is tracking maximum of the 0's of h or j.

And it is also easy to argue the other the other way round right that if z hat is greater than equal to r, which means that at least one of the elements must have satisfied this which means that Y r must be greater than equal to 0 because Y r counts the number of elements that satisfies this.

So, the con so, just negating this is the if and only if. So, just negating both the sides we get this equivalent statement right. And so, this is what this is what they help us analyze this; analyze the success rate of this algorithm.

(Refer Slide Time: 06:51)



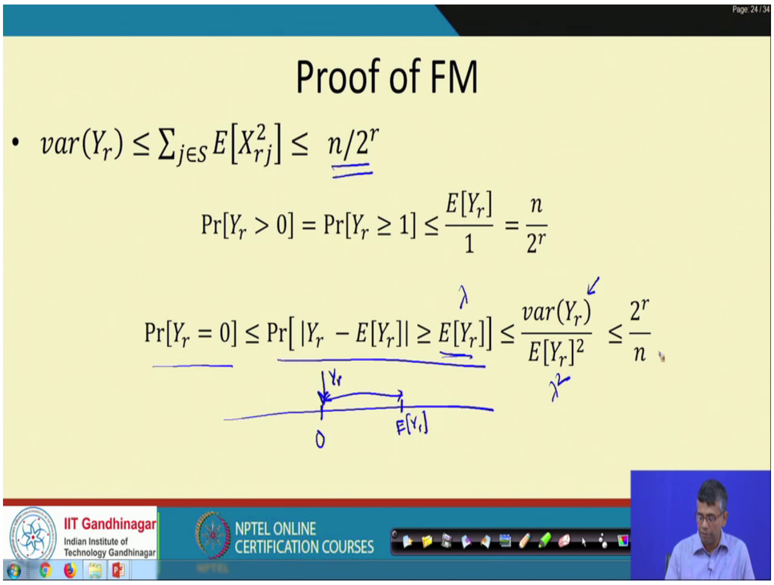
So, let us try to look at the variable Y r. So, we saw that Y r is nothing, but the summation of j belongs to S X r j. So, therefore, the expectation of Y r is again by the linearity of expectation, the sum of the expectations of X r j ok. This is this is what we are saying right and note the distribution of the variables X r j. So, we have said that X r j is 0 1 random variables right, but X r j, when is X r j equal to 1? X r j equal to 1 when 0s of h of j is at least r right and by the intuition and because h of j is a uniformly chosen bit stream right from I mean out of the out of all possible bit streams. The probability that it is divisible by 2 to the r is 1 by 2 to the r, which means at the probability that 0s of h of j is at least r is 1 by 2 to the r right.

So, exactly its one with property 1by 2 to the r and it is 0 else right. So, by plugging in this we can call calculate very simple bit expectation of Y r. So, expectation of Y r is n by 2 to the r because the carnality of S equals n. Also we can say what the variants Y r is because the X r js are i i d because we have.

So, here we are critically assuming the fact that the hash function is independent right or at least will see what will see it is an half to assume that its 2 Y's independent right. In that case the variants of Y r in that case the random variables X r j are such that the variants of their sum equals the sum of the variances right which means at the variants of Y r equals the summation j belongs to s variants of X r j.

Now, again because X r j is Bernoulli random variable. The I can say to the variance of in any case I can say the variance of X r j is less than expectation of X r j square. Because it is a Bernoulli random variable, I can calculate what this is fairly easily a sort will do.

(Refer Slide Time: 09:15)



So, so, the way so, so, the expectation of X r j square right is again nothing, but the expectation of X r j because that X r j is a random variable which is less than or equal to n by 2 to the r. So, now, will apply two inequalities: the first one will be a mark of zero quality, that will bound the probability that that Y r is bigger than 0. And the probability that Y r is bigger than 0 strictly bigger than 0 will be equal to the probability that Y r is greater than equal to 1 because Y r is a integer. And this by Markov's inequalities is the expectation of Y r is divided by 1.

So, if I had lambda here, I were of put lambda here and because and here we using the fact that lambda equal to 1. And this quantity is now equal to the exactly equal to the expectation of Y r which you have calculated to n by 2 to the r. So, that is it for this.

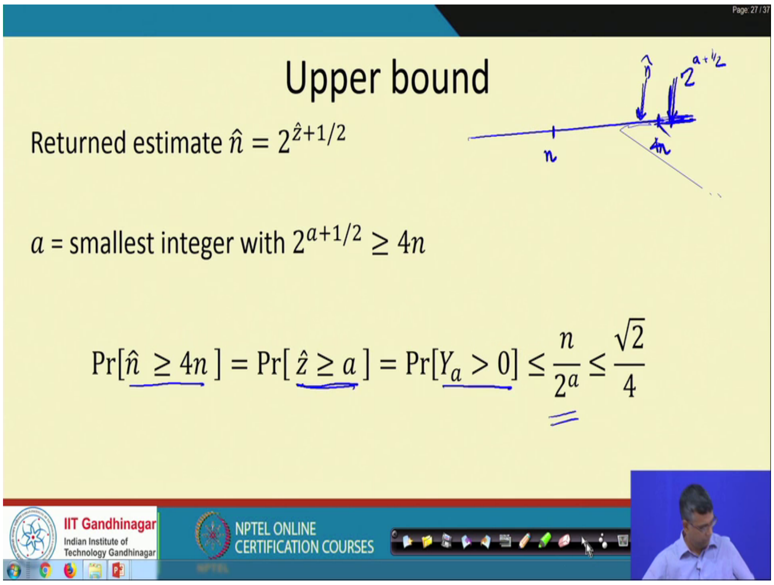
So, now the probability that Y r equal to 0 equal to 0, here we use Chebyshev inequality. We say that, let us look at Y; this 0. Let us look at the expectation of Y r. This is some value here right. Now the probability that remember that Y r does not is never less than 0.

So, the probability that Y r equals 0 right. If Y r does take this value right, then the difference between Y r and its expectation is at least equal to the expectation because the difference between Y r and expectation here at least equal to expectation. So, let us bound the probability of this and we can bound the probability of this to be less than by Chebyshev inequality to be less than equal to variance of Y r divided by; if I had used lambda here, I get to use lambda square here and instead of lambda we are plugging in expectation of Y r.

So, therefore, it is variance of Y r divided by expectation of Y r square. And this and this and because the variance of Y r is also bounded by also is bounded by n by 2 to the r, I get to plug in n by 2 to the r and n by n numerator and n by 2 to the r whole square in a denominator. So, I get 2 to the r by n so.

So, therefore, the two things if you take note is that probability that Y r greater than 0 is that most n by 2 to the r, probability that Y r equal to 0 is at most 2 to the r by n. These are the only two things remember from this slide ok. So, let us see. So, let us proceed with the analysis.

(Refer Slide Time: 11:58)



So, now suppose we have return the estimate, we have because the final value of z or z hat. We have return the estimate equal to n hat equal to 2 to the power z hat plus half that is what algorithm does. Suppose now e is the number; suppose now a suppose this is n.

So, a is the smallest integer such that 2 to the power a plus half is bigger than equal to 4 n.

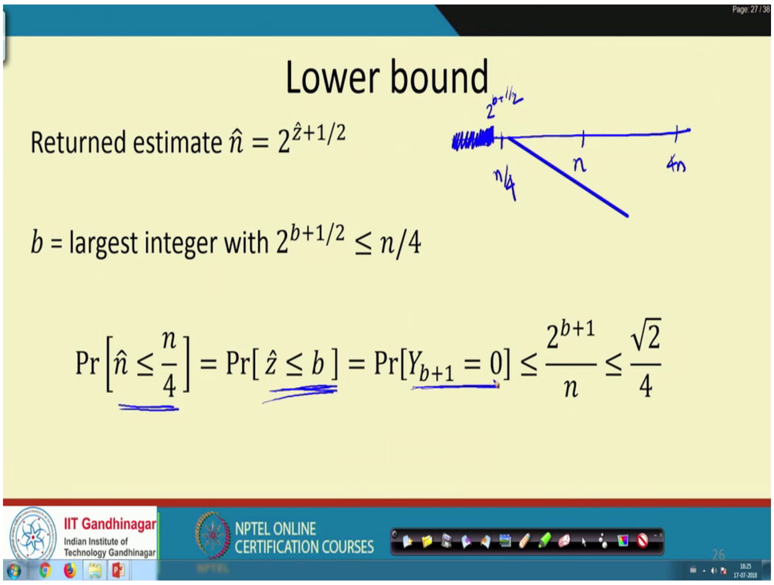
So, this is 4 n here ok; this is 4 n here right and a is the smallest integer such that 2 to the power a plus half just process four n right. So, a is chosen somehow such that 2 to the power a plus half is lies to the right of 4 n ok. So, I want to say that z hat will not be bigger than a and if z hat is not bigger than a which means that z hat lies to the I mean which is that n hat lies to the left of this number, which means that n hat lies somewhere here right. So, n hat the probability n hat crosses 4 n is equal to even that n hat crosses 4 n equal to the z hat is at least a ok.

And let us try to bound the probability of this. So, we have seen that we have seen that event z hat is at least a is same as that event Y a is at least is strictly bigger than 0 because z hat bigger than a means that the I mean there exists at least one element right, with at least one element j with the 0s with the 0s of h of j to be to be bigger than to be bigger than bigger than a and therefore, Ya has to be greater than 0.

But you just calculate it what is the probability of Y is greater than 0 for something right. Probability of Y a greater than 0 is at most n by 2 to the a because of what we were doing here by the uncrossing inequality ok. So, let us plug that in and the and by using and because of the value of a right, you see that n by 2 to the is less than square root 2 by 4. Just plug in this just plug in this just taken by 2 to the by a and just use the inequality that is the value of a that we have assumed out here and you will see that this is less than square root 2 by 4 ok. What does it mean? That means, at we valued at we achieved at holds good. We have said that the probability that to estimate will be bigger than 4 n right.

So,. So, here is n here is n and here is 4 n and we are saying that the probability of estimate lies on this side is at most square root 2 by 4 which is not a very small number, but it certainly less than 1 ok.

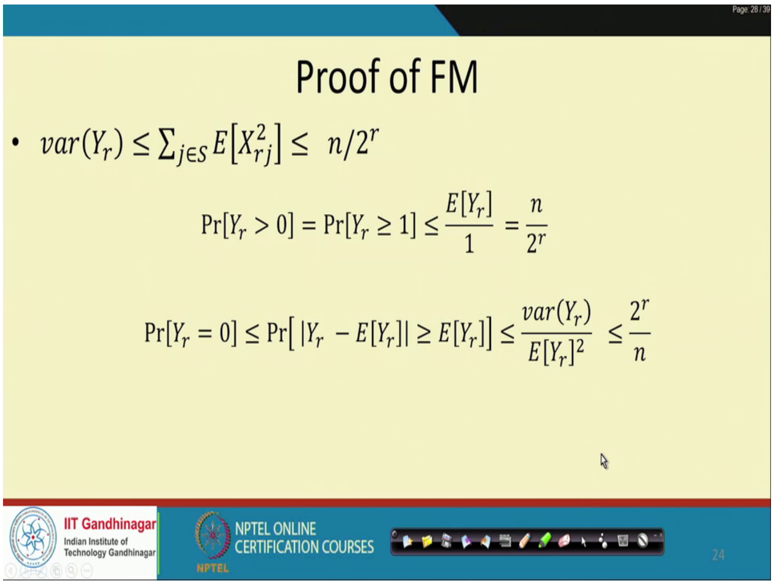
(Refer Slide Time: 15:14)



So, now the other side now we say that ok. Let us look at n we have looked at 4 n and let us look at n by 4 and now what we will see let us define b to be the largest integer to be such that 2 to the power b plus half is less than n by 4 such that. So, basically b, I want to say that b such that 2 to the power is the is such that it is kind of the point out here such that b is the largest integer such that 2 to the power b plus half is still less than n by 4 and I want to say that z hat is not going to be less than b right.

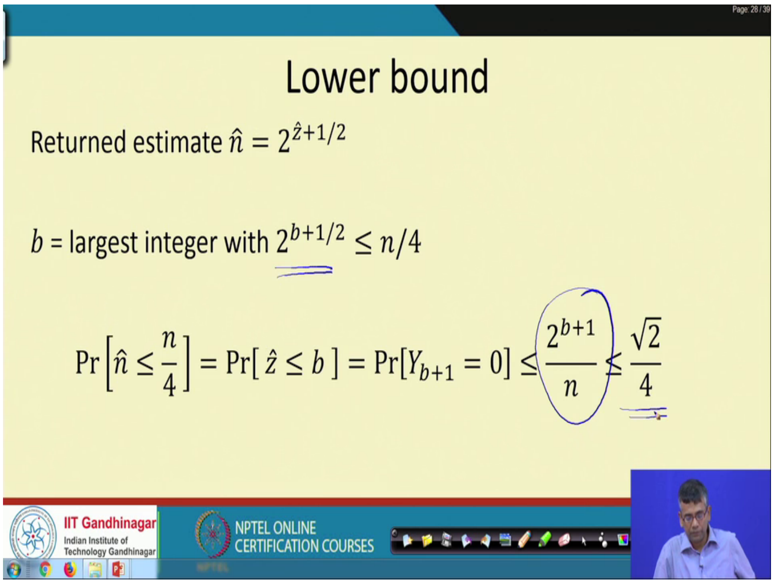
So, z hat is not less than b which means that z hat is at least b which means that the estimated by z hat lies to the right of n by 4 that is what we going to say. So, little more formally probability due to that n hat is less than equal to n by 4 right this event is the same as event z hat is less than equal to b right, but we know that this event is the same as the event Y b plus 1 equal to 0 because Y z hat is less than b z hat is less than b because there was nobody who had 0s who had the functions 0s to evaluate to b plus 1 which means that Y of b plus 1 is equal to 0. And again by sort of using the using the inequality on Y, what we have seen is that.

(Refer Slide Time: 16:51).



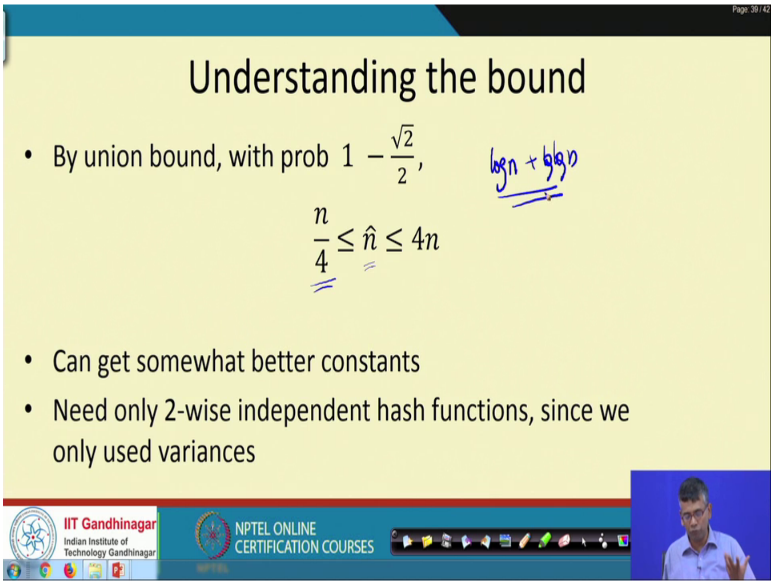
Using inequality of Y we have seen that probability of Y r equal to 0 is less than equal to 2 to the power r by n, then we plug in r equal to b.

(Refer Slide Time: 17:02).



And we plug in r equal to b and therefore, we see that this that this quantity is to 2 to the power b plus 1 by n again we use the definition of b out here and we get this is less than square root P is root 2 by 4 ok.

(Refer Slide Time: 17:32)



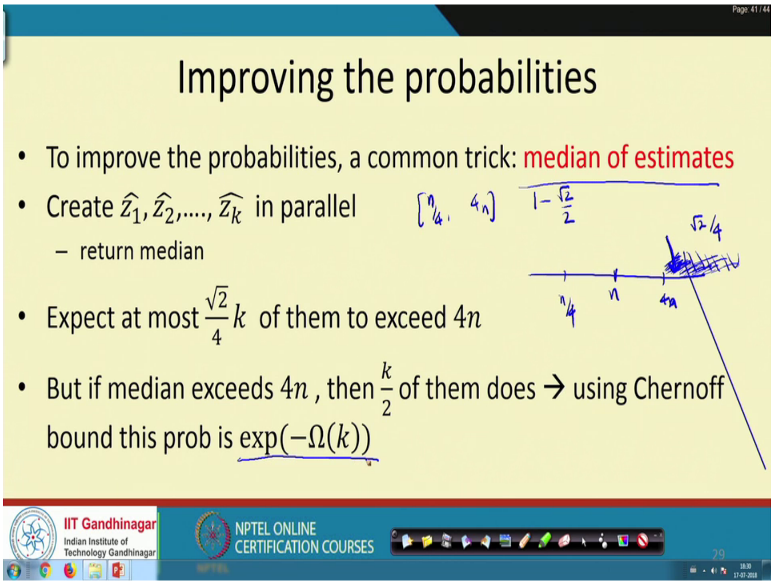
So, therefore, by sort of and therefore, by putting this together and by taking U n bound, we see that with probability 1 minus square root 2 by 2. So, why this called 2 by 2 because we are adding up the failure probabilities square root 4 plus square root 2 by 4 that gives you square root 2 by 2. So, why is square root by 2 or adding up the 2 failure probabilities square root 2 by 4 that gives you square root 2 by 2. So, if logarative 1 minus square root 2 by 2. We get that n hat the estimate n hat lies in the interval n by 4 to 4 n.

So, you might sort of found your eyes in this and you know that, what is the big deal we have we are sort of or estimate might be as bad as 4 times that 4 value and it might sort of b as bad as one fourth the true value is this why is this nice. But remember what we have used only space that is log n plus log, log n right. And we have still given you with very good with the reasonable probability, why? They have still given you an estimate that is spittle tight right. It is within a constant factors, that is not obvious beforehand.

In fact, this is the basic block, we can now improve it quite a bit, we can get some other better constants. So, how can we do that? If you want to look at this exactly, but if you because we look at better algorithms, but if you want to improve the constant 4 out here, you need to use a log base other than 2 right. If you use a log base 1 plus epsilon, you can get some that is related to epsilon here instead of instead of 4.

Now, you also need to note that we did the analysis using independent hash functions; however, we have only used variance nothing else really. I mean we just used for the variance of the some of the variance and that is already true if the hash values are two independent two improvise independents.

(Refer Slide Time: 19:21)



So, that is also important to note then the last point is that if you want to and this is the special trick that you will that you should always remember as a data sign. This is that if you want to improve the probabilities right. You use a common trick; trick name is median of estimates right and this is of very common trick very useful trick that you will that you should use other places also. And if the it is as simple as this that instead of you know that each of the Z i's each of the Z I's gives you an estimate within n by 4 to 4 n with probability 1 minus square root 2 by 2.

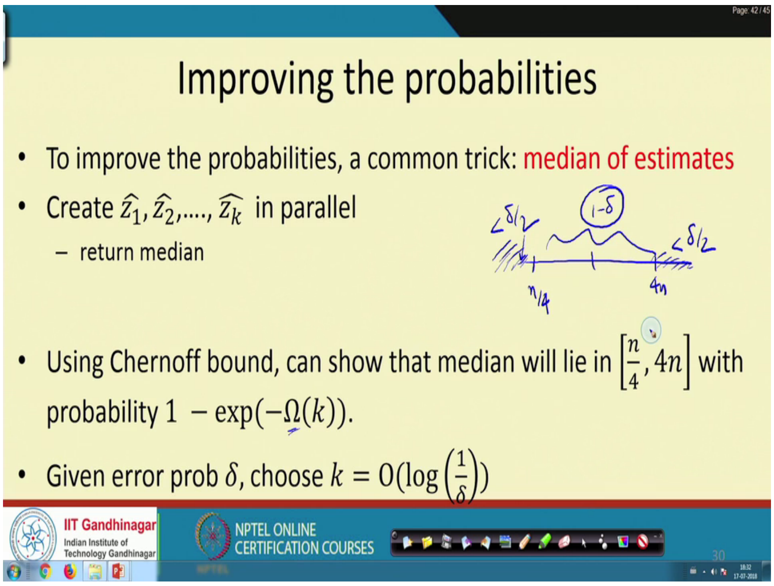
Let us calculate k of them in parallel let us take. Let us let us take k independent hash function that are independent of each other right and then and then calculate k of this estimates right. So, now, instead of one estimates we have k estimates, you return a median of these estimates. Now let us do the analysis for this. Why is this median better right?

So, first of all you note that. So, I mean this was n this was 4 n and this was n by 4. First of all you say that to the right of this the probability that any one of the k lies to the right of 4 n right is only square root 2 by 4. This is what we did right; the probability that it lies to the right in any of the z hat is. Therefore, if I calculate if I calculate k of them right, the probability that any one of them right the I mean the expected number that crosses 4 n expected number that crosses 4 n right is only square root 2 by 4 times k at most right.

Because the probability have been one of the 2 causes for a square root 2 by 4, but then if I calculate I mean, then if I look at the median, then would the median lie on this side of 4 n that is would the median is bigger than 4 n? What happens if the median is bigger than 4 n? The median is bigger than 4 n in at least k by 2 of the estimates lie on this sort of 4 n lie on the y sort of 4 n. If the median is bigger than 4 n, then what is k by 2 of estimates is? Bigger than 4 n right.

And by using chain of bound what you can say is that this is very unlikely that I expected only square root 2 by 4 k of them to exit 4 n, but k by 2 what is the chain k by 2 of them really does. And using the chain of bounds and you are not going with the details of it. You can show that this probability is exp of some constant times k exp of minus some constant times k.

(Refer Slide Time: 21:59)

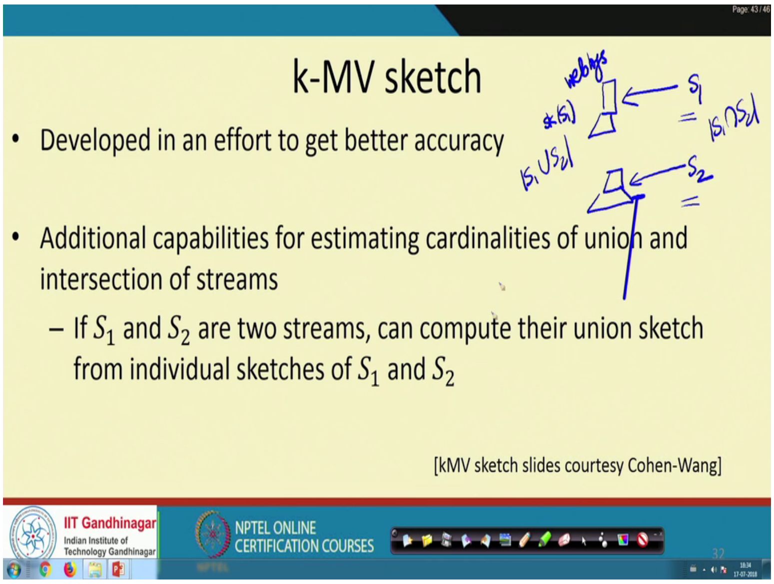


So, therefore, therefore, by sort of to get a error probability of delta right, you choose a to b log 1 over delta and what you can see is that with the probability that the median lies to the right of 4 n right is less than let us say delta by 2. The probability the median lies to the left of n by 4 is also less than delta by 2 right.

And therefore, the median with probability 1 minus delta, the median lie must lie in this range n by 2 by 4 n ok. So, we need to fix in the actual constants here in this in this omega and I have sort of told you over the actual constants are. And it would be a good exercise for you to sort of figure, how to actually plug in Chernoff bound actually to see what the constants are and you get to see convinces whether the constants are actually indeed very small in theory right. It is like something like six probability is enough.

So, we also dint do the analysis for the left hand side, but it is exactly the same as before that if the median lies to the left of n by 4, then at least k by 2 of them lie to left of m by 4 and that is really unlikely. So, what it shows what it tells me is that once I have the basic building block of the Flajolet-Martin that is giving a constant factor of approximation with the constant probability. I can get the same constant factor of approximation guaranteeing the same constant factor approximation with the very high probability and this is going to be a very useful technique in practice.

(Refer Slide Time: 23:24).



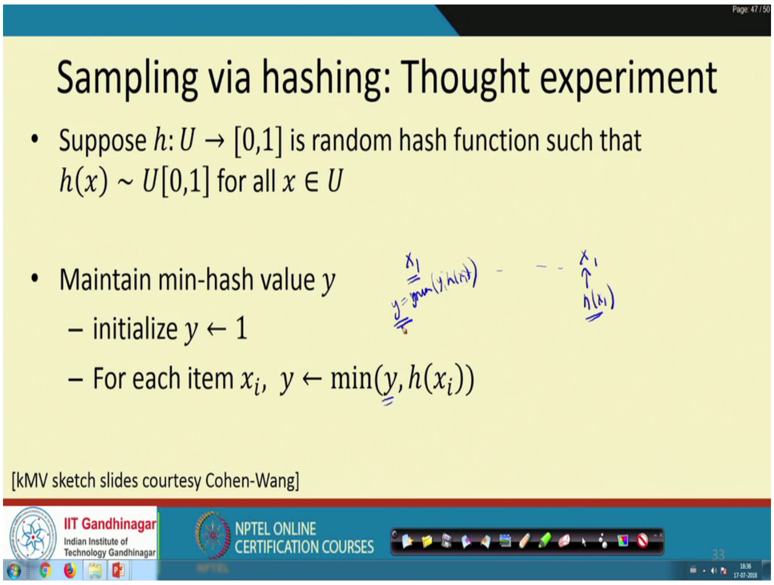
So, this is good, but we will do even better. In fact, we look at a sketch that number 1 will give us a slight better accuracy right. And number 2 it will also have some additional capabilities and this will be very interesting. The additional capabilities are as follows that suppose we have one machine here right that is getting a stream S 1; you have another machine here that is getting a stream S 2.

So, imagine this as follows that this are that this is your cloud and your cloud and this is your web logs. So, the machines are calculating the logs of your users right. And this machine somehow gets the log of the first day, this machine sum of the bits of log of this second bit right. Now; obviously, the set of users from day to day, they might differ. They do not differ by so, much. So, what you might going to say is that, what is the value of S 1 union S 2? What is the value of S 1 intersection S 2 ok? And here is what will be able to do? What will do is that they will create a sketch of S 1 here. It will be a data structure that we calculate here.

So, the naive way of doing this be to just shift entire data S 1 to the second day to the second shim of the entire S 2 to the power first machine. That might be a whole lot of data, but here is what we will do. Suppose we could create this sketches that we are creating right which are very small data structures. So, remember the Flajolet-Martin was size only logarithmic in the size S 1. Suppose we could create this and then we could ship it to the second issue or we could create it in a store and we could ship it there. Then the amount of communication that we need will be really small. And we can we enable this? It is not entirely clear how to do this with the Flajolet-Martin ok. So, although it is not very hard either, but, but here is what will do? But will create is this is this kMV sketch that will allow us to do unions as well as intersections and and this will be very very useful.

So, by the way a lot of the slides that we have used here that we are using here are derived from this from this set of slides by Cohen and Wang and we will provide you pointers to these also.

(Refer Slide Time: 25:57).

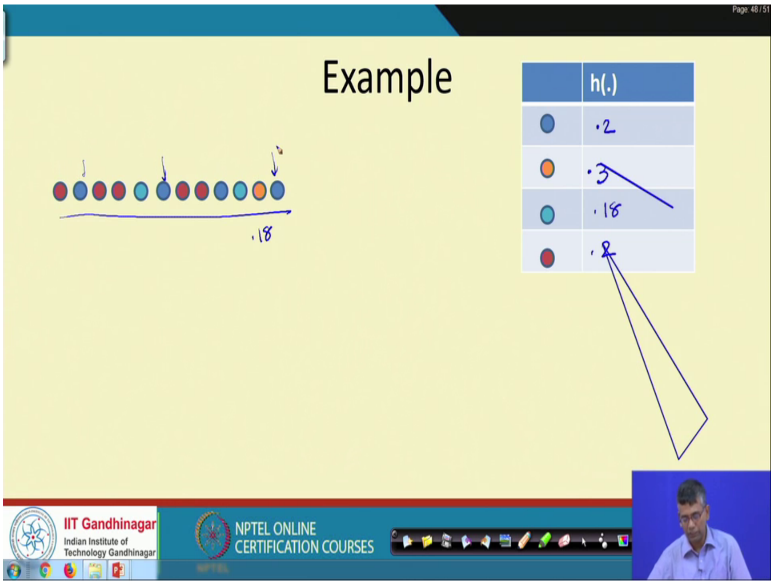


So, first here is a thought experiment right. So, imagine you have a hash function that maps from a universe to numbers between 0 and 1 right. I mean until now we have been only been mapping into integers or bit streams where you might as well map in to into the floating point numbers into the real numbers from 0 to 1. And suppose this hash function is random and is such that that h of x is a uniformly chosen number between 0 and 1 right. So, it is the uniform distribution. So, h of x is a random variable that satisfies the uniform distribution for every x you pick according to the uniform distribution you consider 0 and 1.

So, suppose we do this, when we when we see the elements x. Suppose we look at the at the minimum of the hash values that we have seen. So, just like 0s that we had looking at there; suppose we look at the minimum hash values that we have seen until now. So, we start, but initializing y to be 1 and for each item x i we said y to be min of y and h of x i right and let us keep on doing this. So, why is this useful? Couple of things hit you suddenly, then suppose you see x 1 here and you see x 1 here later right. So, this y already has done a y equal to min of y and h of x 1 here right. So, therefore, when it sees x 1 here h of x 1 is a same value right and therefore, this x 1 is not going to have any effect on this y right. It is not going to decrease this y any further right because y is already less than h of a h of x 1 ok.

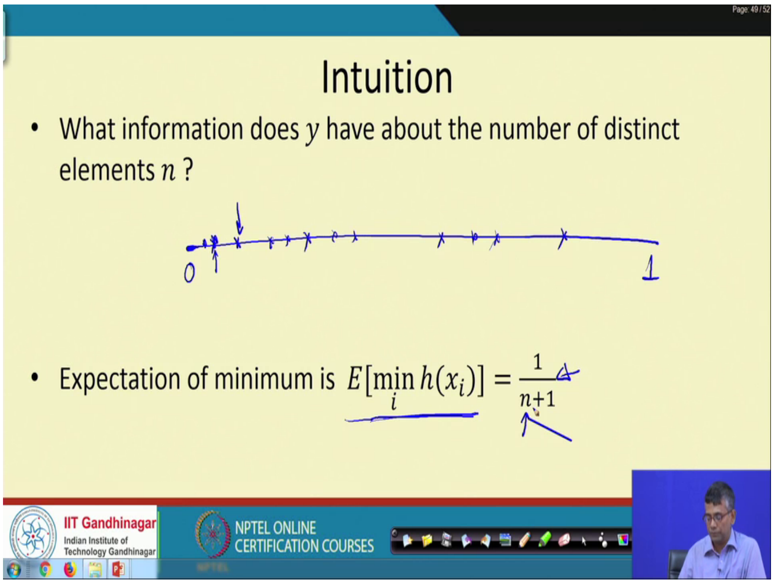
So, that is very useful right.

(Refer Slide Time: 27:52)



So, let us look at a specific example for instance and supposing h has maps this blue element in to 0.2 maps, this orange element in to 0.3, maps this into 0.18, maps this into 0.2 ok. So, it is not very hard to see that when you go with this stream right, the h of x 1 will just be 0.18 right because just it just calculates a minimum of these values right. And fag that an element is appeared multiple times is not going to have any effect.

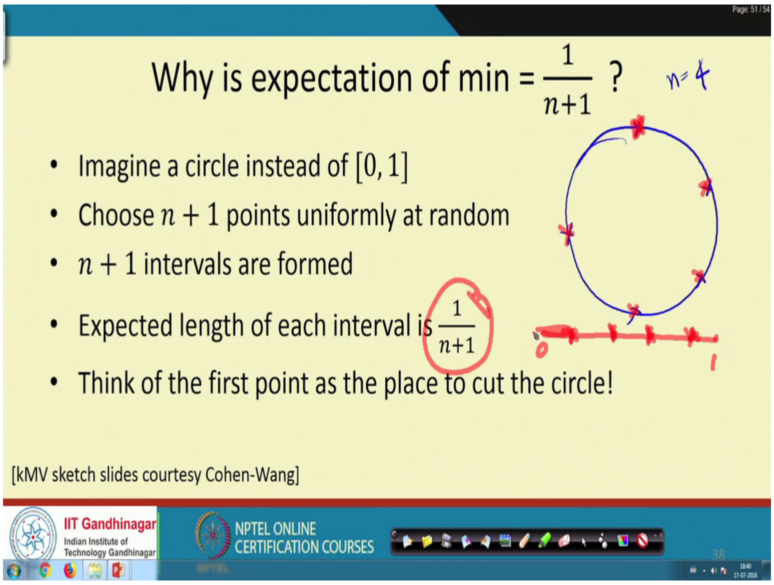
(Refer Slide Time: 28:31)



So, but then so what? What is the; what information does this minimum value have on the number of distinct elements? So, for this imagine an a imagine this process. So, h x for each x there is a h of x this is the value between 0 and 1; for each x there is a value of x; for each x there is the value of x for each x there is the value of x right.

So, if you keep on throwing more x right and we are always tracking the minimum right; so, if you keep on throwing more x this minimum will decrease right the value of this minimum decreases as you keep on throwing more x. And in fact, what you can say is this very interesting statement that if you trapped the h of x i the minimum value of our i h of x i the expectation of this is actually exactly equal to 1 over n plus 1, where n is the number of distinct elements ok. Expectation of the minimum is equal to 1 over n plus 1. This seems varying sort of magical again right. So, why is this true? Let us let us see let us kind of see a picture prove of this.

(Refer Slide Time: 29:44)

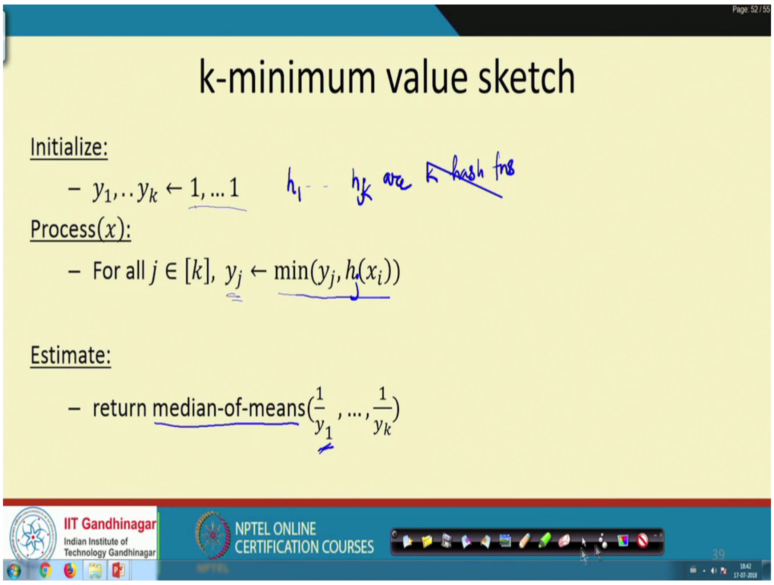


So, supposing imagine a circle. So, we were we were actually dealing with the interval 0, 1, but let us imagine a circle. So, let us put in the circle and now choose n points in the circle uniformly at random. So, let me choose like 3 4 5 points in the circle uniformly at random right; suppose so, n equal to 4.

So, now just pretend that any one of so, now, because am throwing I mean n plus points there are n plus 1 intervals 1 2 3 4 5 intervals. Because of n plus 1 is 5 right. So, the expected length of which interval is exactly 1 over n plus 1 right; expected length of h interval is exactly on 1 over n plus 1. Because all the intervals are equally I mean there is nothing distinguishing one from the other. So, now, that is it think of the first point that you have thrown as where you cut the circle and if you cut the circle and flatten it out, you just get the interval from 0 to 1 right. And then and then each position out here corresponds to one of these stars in the one of these.

So, each of these other points that we have thrown down corresponds to one of the points in interval 0 to 1 right and because we and because we argued that expectation of each interval is 1 over is 1 over n plus expected length which interval is 1 over n plus 1. The expected length of this one right which is the expectation of the minimum of the hi is also 1 over m plus 1 right and that is it ok. So, this is theoretic varying it and this process courtesy Cohen and Wang.

(Refer Slide Time: 31:36)

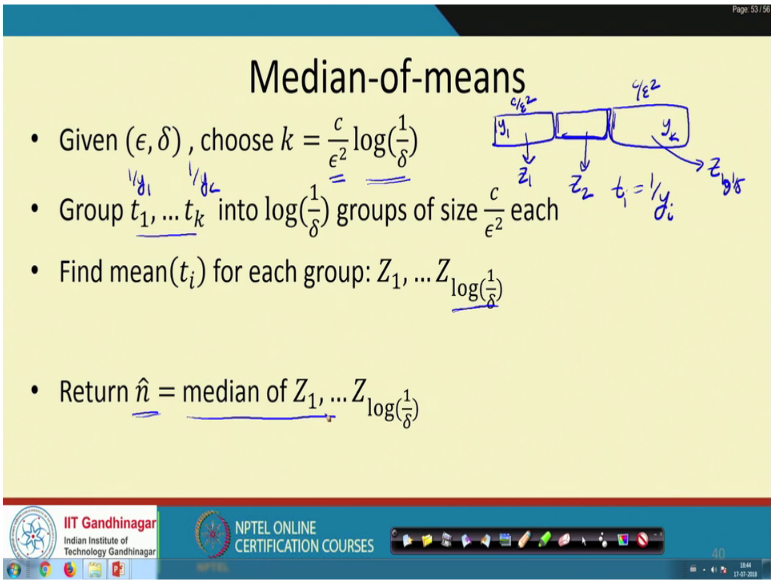


Now, the it appears in other places in the literature also ok. So, here is my sketch. We all going have k values what I want to y k, rest of them are initialized as 1 1 1 1 because 1 is the maximum value it can take. Now when you as you process the stream when the element x comes for each j for each of the j from 1 to k, I will set that y j to be minimum of y j and h of x value basically y j will track the minimum. So, this should really be h of j.

So, there should be j hash functions h 1 to h j. So, y j should track the minimum of the j th hash function; h 1 to h 1 to h k are k hash functions and y j will track the minimum of the jth hash function.

And now, the for the estimate you call will do something a little smarter than what we were doing before remember. What we really supposed to do is just you because see the expectation is 1 over n plus 1 expectation of y i. So, therefore, if I just involve it if I just do 1 by y i right, I should get something that is close to n right we do not exactly do that. Instead what will take this balance 1 by 1 1 by y 2 1 by y k we believe that each of them have expectation close to n, but will call a procedure called median of means ok. I will just tell you the moment for medium of means is and this is another one of these tricks that to should learn as a data scientist.

(Refer Slide Time: 33:31)

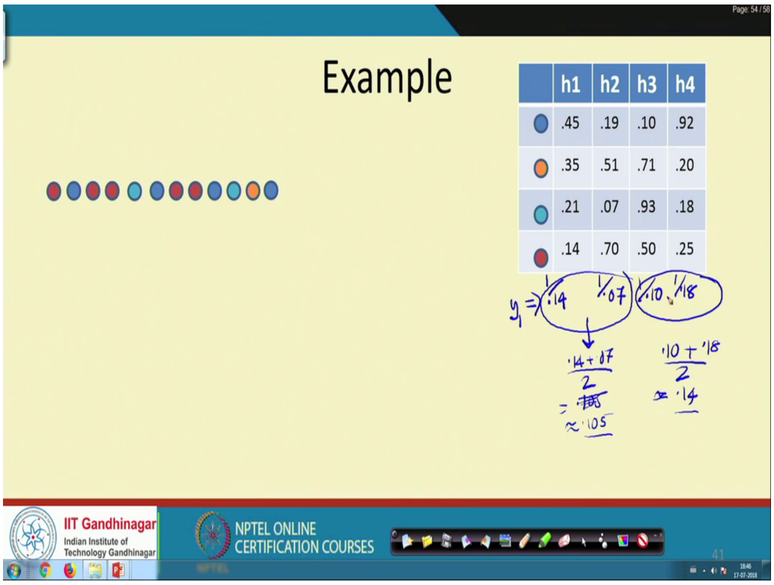


So, here is the median of means. So, suppose you have; suppose I choose the value of k to be like some constants c by epsilon square log 1 over delta. Basically think of think of y 1 to y k y 1 to y k has been partitioned in to blocks. Each of this blocks one of each of blocks will of size c by epsilon square. So, epsilon and delta are things we sort of talk about little bit in a bit, but in intuitively they relate to the epsilon delta approximations that we have been talking about until now. So, epsilon is going to be the error parameter and delta going to be my confidence.

So, will set so, will start out with k to be c by epsilon square log 1 of the delta. And after we have obtained estimates of the y is will route them into groups of size c by epsilon square. So, there will be log of 1 over delta groups such groups right. And suppose and suppose so, so call t i to be 1 by y i ok.

So t i to be 1 by y i and then we are grouping these into log of 1 over delta groups of epsilon c by epsilon square each. So, each of the groups find out the mean of that group right; so, for a first group find out the mean z 1, for the second group find out the mean z 2 from the log 1 over delta group find out the main z log 1 over delta. So, there are this many this many of the means. Then return the median of the means of these means. So, first we chunk the estimator into in to certain number of groups, we find out the means of the groups means of the averages of these groups right. And then we return the billions of these groups and this turns out to be a very globastic estimator right and let us see y right ok.

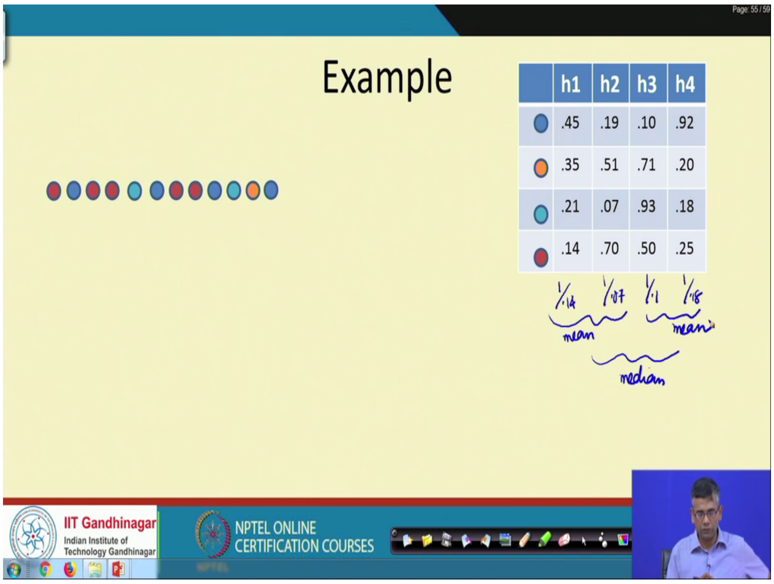
(Refer Slide Time: 35:31)



So, so let us kind of runs one example. So, supposing now we have food hash functions and we and now and now we say that each of the hash functions map the values in the range from 0 to 1. So, h 1 maps each of these each of these elements into these values. The it maps the blue element into 0.45, the yellow element into 0.35 turquois into 0.21 and red into 0.14 and so, on. So, now, according to h 1 you will calculate estimated y 1 right which will be equal to the mean of the mean of all these values all the h 1 values over all the elements that you have seen. And in this case it will be equal to 0.14. So, y 2 will equal mean of all these values. So, 0.07 y 3 will equal then mean of this values which is 0.10, because all these elements have appeared in the stream and y 4 equals the mean of this values to be 0.18. Now let us say we group them into groups of size 2 right.

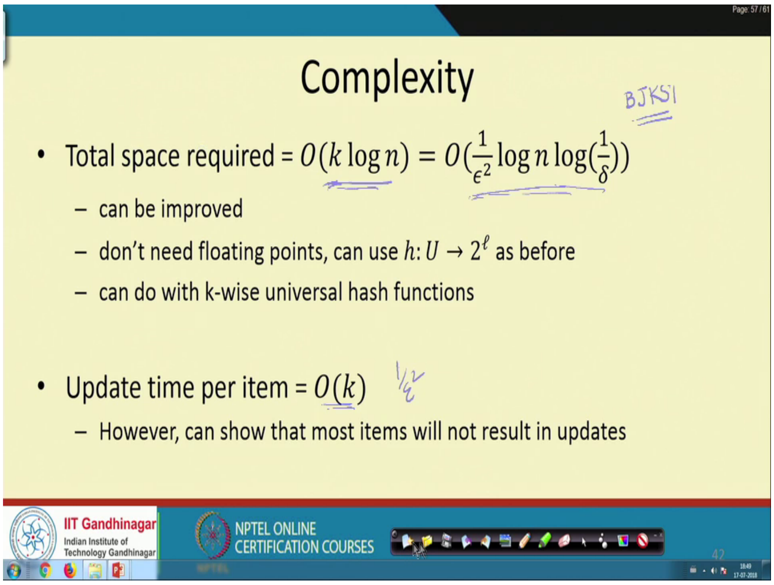
Then we calculate their mean. So, 0.1 4plus 0.07 plus by 2 0.10 plus 0.18 by 2 which is 0.28, 0.14 which is 0.21 0.10 something like 0.105; now point 0.1 0.21 which is something like 0.105 and so and so, you take then you take the median of this. So, at this point for this example taking a median does not mean much because there is only 2 such estimators, but if you imagine multiple such estimator and you will take the median of these. And then and then and then you will return the, well actually I mean I made a slight mistake you need to take the 1 over of these 1 over of these and then return the means of and then calculate means of these.

(Refer Slide Time: 37:37)



So, it would not actually be like this it will be. It will be I means the first estimator will be 1 over 0.14 second estimator will be 1 by 0.07. Third estimator will 1 by 0.14 th estimator will be 0.1 by 0.8 and then first you calculate the mean of this and then mean of this and then the then the median of this ok.

(Refer Slide Time: 38:04)



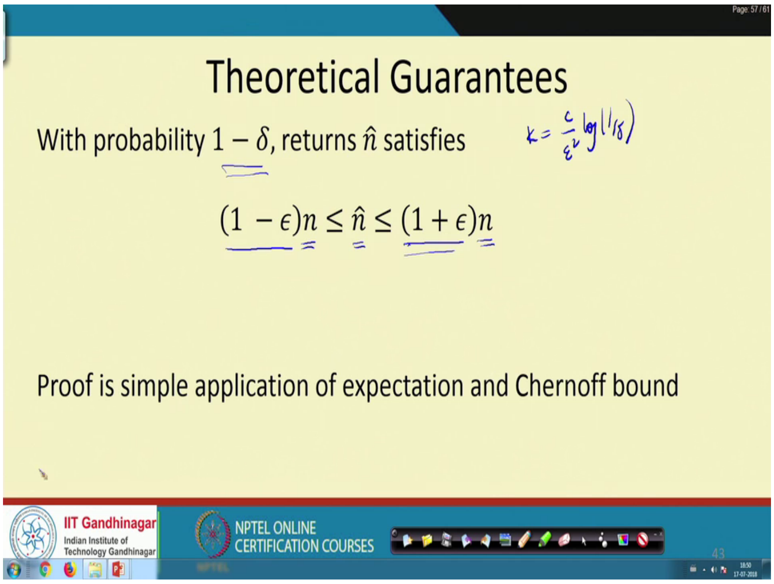
So, let us calculate first the first the again the space required the complexity of it. So, we are keeping k estimators right and do I keeping real numbers what is the complexity of that? I do not know, but it is not again very hard to show that you need to keep a real numbers only up till length some c log n right just like using birthday paradox we were analyzing that.

If we have if we map them in to universes of size c log m there would not be any collision I mean 2 elements with very high probability, probability two elements would not map into the same to the same value of h of x. You can you can again sort of justify here that if you keep a precision of c log m right that is enough right.

Therefore, each of these real numbers each I mean it is enough to keep each of the I mean each of the h of each of the target values of h to be to have a precision of 2 log m. Therefore, the total space is required k times c log n and we have chosen k to be 1 over epsilon log 1 over delta. Therefore, that that total space is 1 over epsilon square log 1 over delta and this can be improved right it can be it does not it.

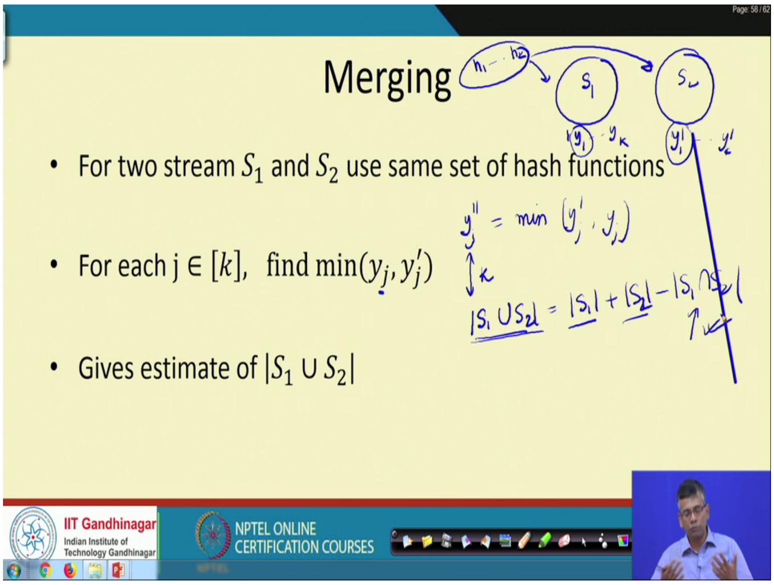
So, so, this algorithm was one suggested by Bard Joseph, Jairam, ra Tavikumar, Shivakumar and BJKST last thing I have forgotten. We can also deal with k y is universal hash functions. They do not need to be completely at random one sort of slide number is that the update time per item is order k and k is scales one over epsilon square. However, we which means that for each item you need to spend one of epsilon square time in updating; however, what you can show is that most items will not result in update. It is not very hard to show this either right which means so, the total amount of time is spending updates is fair is fairly small.

(Refer Slide Time: 40:11)



And then final guarantee right putting everything together is as follows that if you have chosen k to be something like c by epsilon square log 1 over delta right, then using fairly simple sort of application of expectation and chain of bound what you can say is that the estimator the n hat lies within 1 minus epsilon of the true value n and 1 plus epsilon of the true value of n. It lies in this interval with probability 1 minus delta. So, the proof of this are not doing I mean again not doing the proof of this complete proof of this. You can find this in the reference given in over the page course of page as well as the at the end of this slides. It is a fairly simple application of the expectation and Chernoff expectation, variants and Chebyshevs bound and then finally, a Chernoff bound right. It is a something that we have touched in bound before all of them all of the components.

(Refer Slide Time: 41:15)

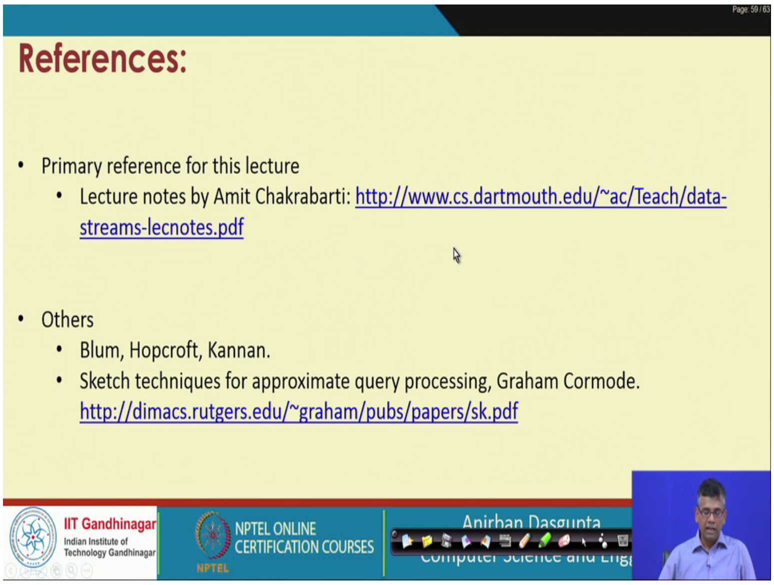


So, before so before I end, so here is the here is the particular example that I was talking about. Here is a particular application where I was talking about that. Suppose we have 2 streams S 1 and S 2 that are disjoined. So, S 1 lies in n 1 machine S 2 lies in 1 machine right and, but they, but there over using universe right. So, I want to know what is the carnality of union of S 1 S 2 or what is the carnality of intersection of S 1 S 2. So, what do we do right?

So, that is a very simple thing you can do first make sure that both S 1 and S 2 that you calculate you run k union both S 1 and S 2 using the same hash functions. So, you choose the same hash functions h 1 to h k communicate that to both the machines right. Suppose this gives values y 1 to y k these gives values y 1 prime to y k prime right and then you add them up you take minimum of y 1 and y 1 prime minimum of y 2 and y 2 prime, y 3 prime and so on right.

So, now you have y j double prime to be equal to minimum of y j prime and y j right. Now you have k such estimates and I use just use this and that is your and that is your sketch and this is exactly a sketch for the union of the 2 sets right. So, it satisfies in some sense a linear property. So, you can say you can convert unions into taking the minimum and see here is one very interesting thing if you can estimate using inclusion exclusion principle if you can estimate the mid the union you going to estimate the intersection also right eh. If I have an estimate of this, I have an estimate of this; I have an estimate of this and have an estimate of this. And in fact, it done said you can do slightly smarter things right, you can also take intersections of the values right. They would not would not set of good there that, but that is something that is sort of an area factor research and you can provide a point us to such papers, if you are interested.

(Refer Slide Time: 4323)



So, finally, here are the references for this for this lecture. The primary reference for this lecture is this lecture notes by Professor Amit Chakrabarti at Dartmouth an excellent set of lecture notes. There is also book on online available book by Blum, John Hopcroft and Ravi Kanaan that will put up in the course that have been that have been web page as well as a set of monograph or monograph by Graham Cormode and lot of the slides are also derived from the science programs as well as the slides Cohen and Wang.

So thank you.