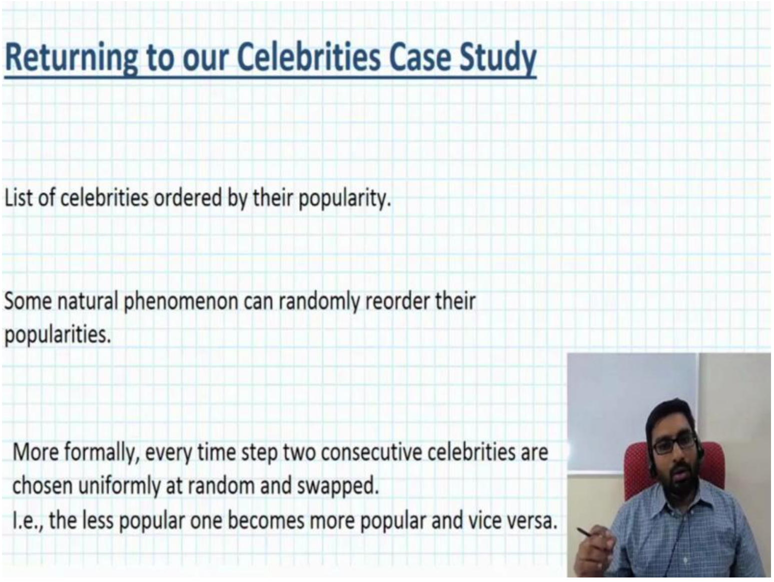
Algorithms for Big Data
Prof. John Ebenezer Augustine
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 18
Returning to our Celebrities Case Study

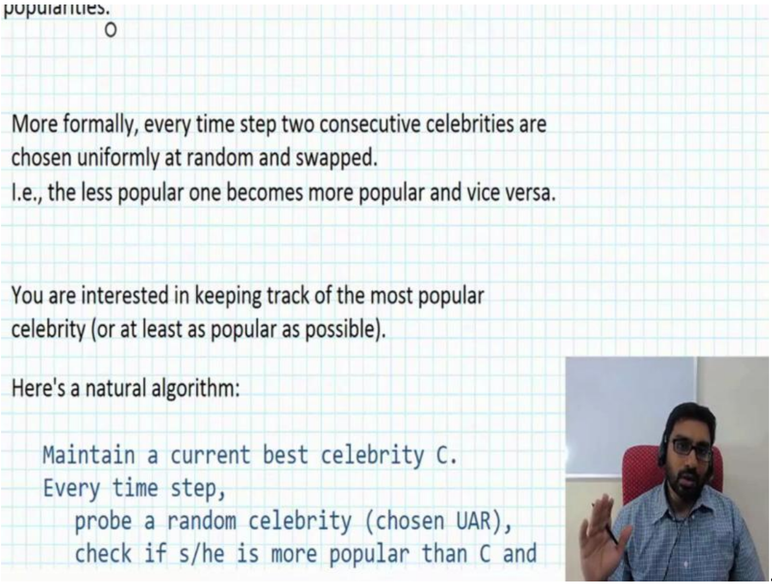
(Refer Slide Time: 00:13)



In the previous segment, we looked at stationary distribution and we looked at different ways to compute the stationary distribution. And, we motivate that this study of stationary distribution with a problem drawn from big data applications. I am going to revisit that problem. I am going to apply the notions that we have studied, in order to be able to solve that problem.

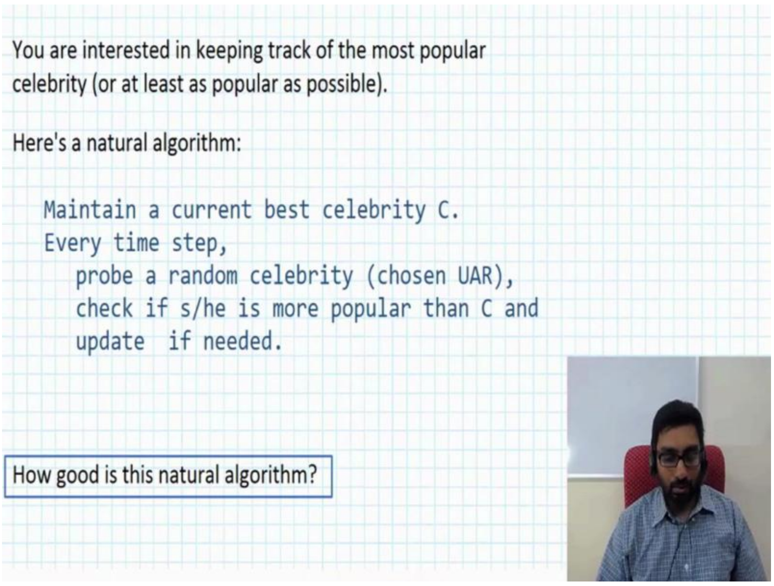
So, just to remind ourselves we were given a list of celebrities. And, they are ordered according to their popularity. There is some natural phenomenon at work by which these celebrities, their popularities get reordered one step at a time. So this is a very dynamically changing order.

(Refer Slide Time: 01:04)



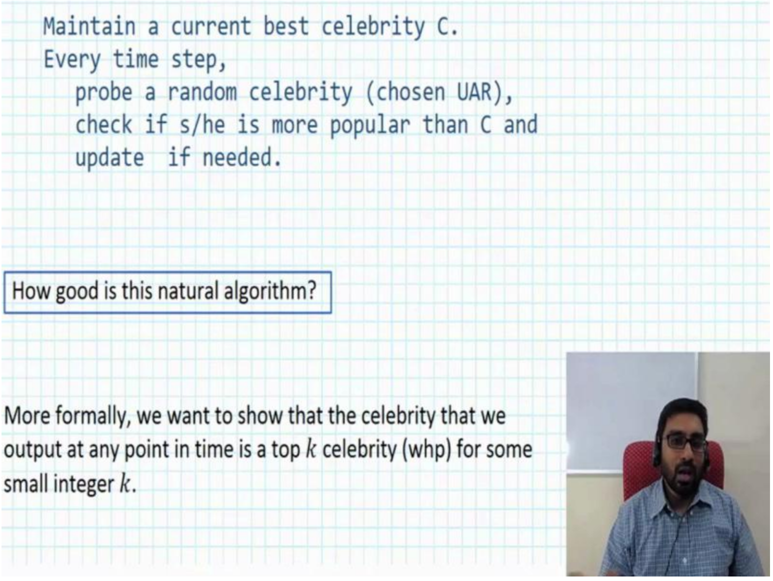
And more formally, every time step two consecutive celebrities are chosen uniformly at random and their popularities are swapped. So, the less popular one becomes the more popular one and vice versa. And, we are interested in keeping track of the most popular celebrity.

(Refer Slide Time: 01:22)



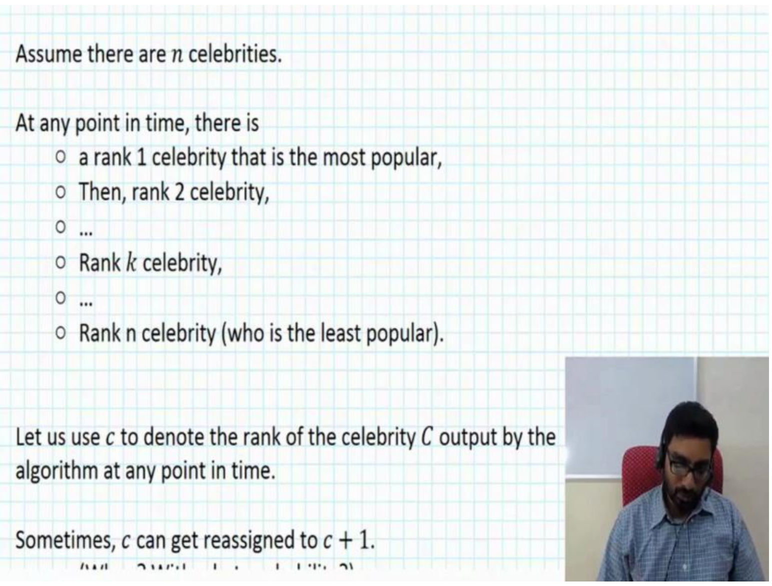
And, we have looked at a very simple algorithm. We maintain the best celebrity C. And every time step, we probe a random celebrity and check with the celebrity C. And, if there is an improvement in the random probed celebrity, then we update C. A natural question refers how good is this algorithm? This is nice. This is a very simple algorithm. It does not have much of an overhead, but we want to know how good this algorithm is.

(Refer Slide Time: 01:54)



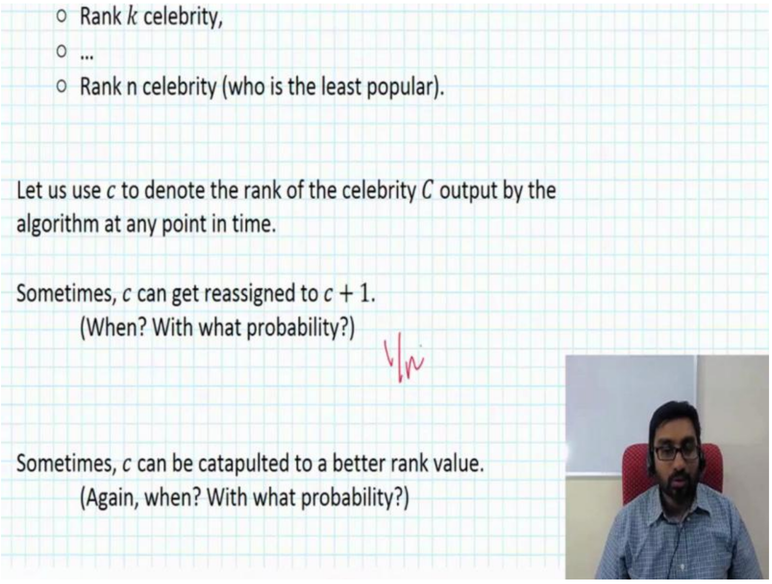
In particular, we want to find a value k with which we can make this guarantee. We want to show that the celebrity that we output any point in time is a top k celebrity with high probability; meaning, probability of the form 1 minus 1 over n for some small integer k. So, we want to be able to say that well, if we output a celebrity with high probability that celebrity is going to be in the top k of the list.

(Refer Slide Time: 02:28)



We are going to assume that there are n celebrities. And just to be clear, the celebrities are ranked from rank one being the most popular to rank n being the least popular. And, let that use the letter c to denote the rank of the celebrity, capital C, output by the algorithm at any point in time. So, the c keeps changing. So, it is quite natural to view this as some sort of a stochastic process.

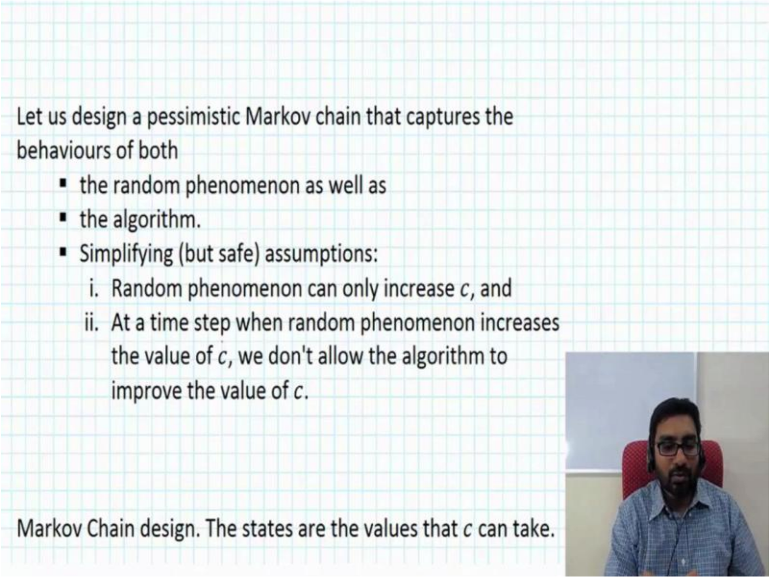
(Refer Slide Time: 03:08)



It is in some cases; this c can be reassigned to c plus 1. When will this happen? Well, if c gets reassigned to c plus one; that means, the celebrity that we are outputting has actually gone down the list. This has become more; this has become less popular. When will that happen? Well, it will happen when the natural phenomenon chooses the celebrity c and the next less popular celebrity. And, then their popularity is what swapped.

So, then this happens with probability something like one over n. Sometimes, however we can get lucky and c can be catapulted to a better rank value. So, it can be improved something from 1 to c minus 1. When will that happen? When we probe or we do a random probe every time step, if that randomly probed element celebrity has a much higher popularity status or is a more popular celebrity, then this sort of an improvement will be seen.

(Refer Slide Time: 04:21)

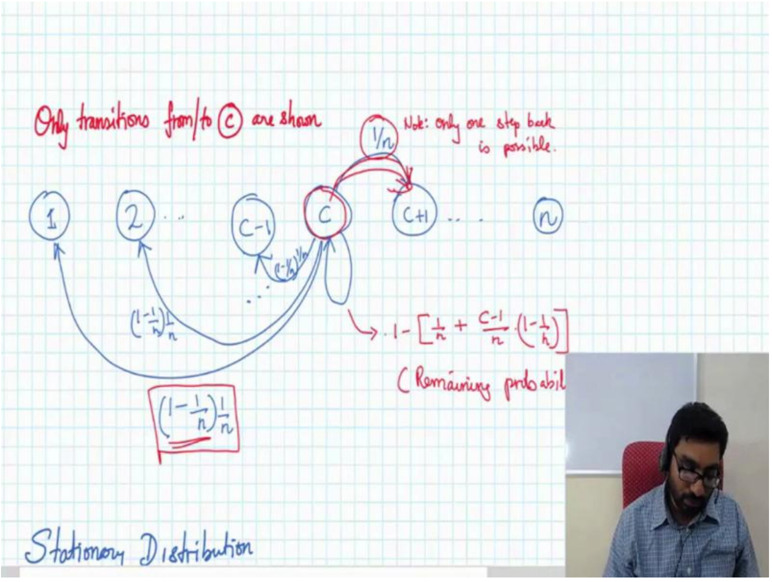


So, what we are going to do is we are going to design a Markov chain that captures these notions. And, we are going to be a bit pessimistic. And, we are going to design the Markov chain, so that it does not exactly capture what the algorithm does. But, in the sense that if this Markov chain, we can claim something about this Markov chain, something good about this Markov chain, then the same thing can also be claimed about the algorithm. And, this Markov chain has to capture two things. It has to capture both the random phenomenon by which your c value can increase to c plus 1 as well as the algorithms effects. There by a randomly probed celebrity is being much better. And therefore, the c value gets catapulted to much better value.

We are going to make some simplifying assumptions. This is where we are going to be a bit pessimistic, but nevertheless simplifying, but safe assumptions. We are going to assume that the random phenomenon can only increase the value of c's. That is, it can only affect us in the negative way, where c becomes c plus 1; because the random phenomenon can also help us. But, we are simply going to ignore that. And, at any time step when the phenomenon, the random phenomenon, succeeds in making c go from c to c plus 1, there is also the possibility that the algorithm may be able to correct that. Not only correct, it can be actually improving the c to a much better value. But, we are simply not going to allow that to happen.

In that sense we are going to simplify the Markov chain, where if the random phenomenon, underlying popularity phenomenon swaps c and c plus 1. Then, there is no improvement that we see. Under these assumptions, we are going to design the Markov chain.

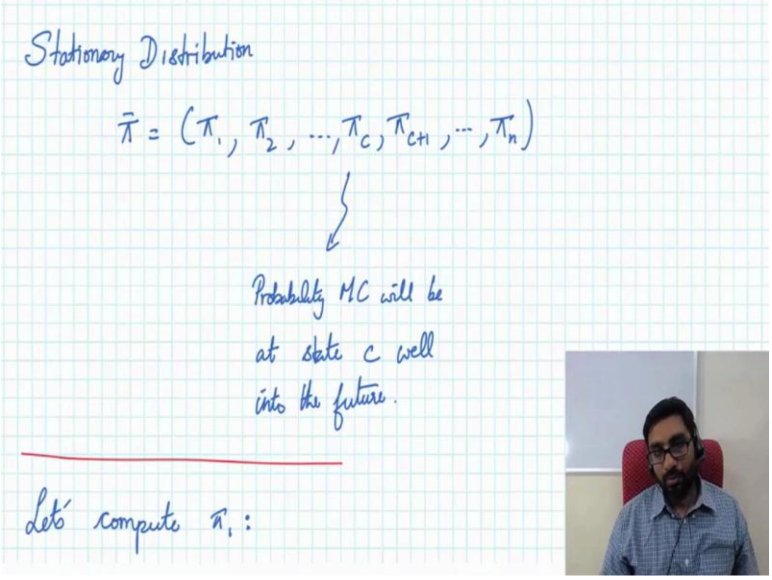
(Refer Slide Time: 06:29)



And, the states of the Markov chain are simply going to be the values that c can take. Of course, if the c can take 1; which means, it is the highest ranked celebrity. And, it can take the value n; which means that it is the lowest. And, it is pointing to; capital C is pointing to the lowest ranked celebrity. In this Markov chain diagram we are only showing the transitions from and to c. So that, its many transitions are not showing.

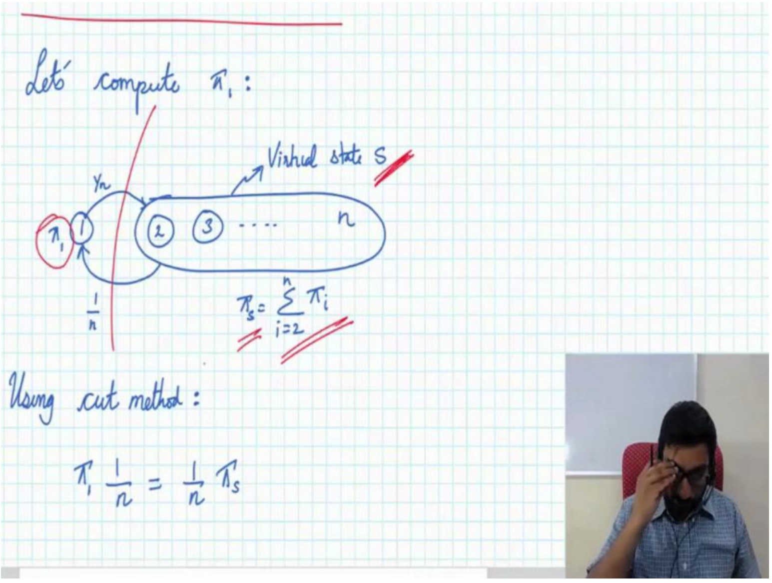
If you are at c, when the random phenomenon affects you, you go one step backwards. And that happens with probability 1 over n. And, alternatively you can be catapulted to a better position. And that will; let us look at the probability with which you will go from c all the way to state 1. That is going to happen with probability 1 minus 1 over n times 1 over n. Why is that? Well, the possibility of making any step forward happens with probability 1 minus 1 over n; because with 1 over n probability, you are going to move backwards. Given that you are not moving backwards, that is this 1 minus 1 over n probability, and then you move all the way to 1 with probability 1 over n. Now, with this same probability we will also be able to transition from c to 2 and c to 3 and so on, up to c to c minus 1.

(Refer Slide Time: 08:19)



(Refer Time: 08:20) the stationary distribution, the vector pi to be pi 1, pi 2 and so on up to pi c, pi c plus 1 and so on up to pi n. And, of course pi c will be that the probability that the Markov chain will be at state c, well into the future 1. So, we are interested in the steady state behavior of the algorithm. And that is what a pi c will be the probability with which the Markov chain will be at state c. So, let us compute this last stationary distribution.

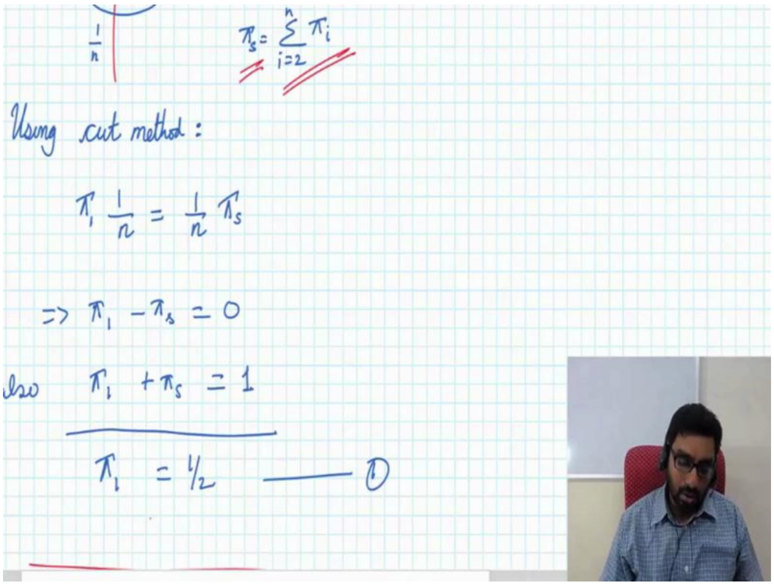
(Refer Slide Time: 09:01)



Let us start by computing pi 1. Pi 1, we are going to compute that by looking at the cut between state one and the rest of the states. So, the rest of the states we are going to call that a single virtual state. As and if you are looked with, when we are at this, when we viewed this way from this virtual state S with probability 1 over n, let us begin by computing pi 1. In order to compute pi 1, let us consider the cut between state 1 and rest of the states.

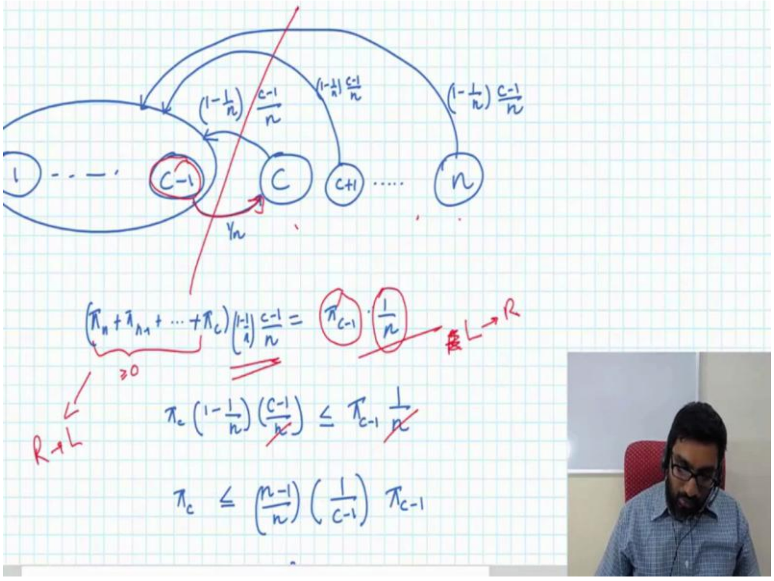
Now, the rest of the states form a virtual single virtual state S. And, we also; in that sense, this Markov chain looks like the two state Markov chain. That we have already looked at in the previous segment. So, pi 1 is the stationary distribution of being at state 1 and a pi S is the probability at which will be with any of the other states. And that is equal to the just the summation of the other probabilities. And, if you are in state 1 with probability 1 over n, we will be transitioning to state 2, which is essentially virtual state S. And, if we are in any of the states in the virtual state, then the probability 1 over n, we will be transitioning to pi 1.

(Refer Slide Time: 10:52)



So, simply using the cut method we will get pi 1 equals to a half. And, this is something you can verify.

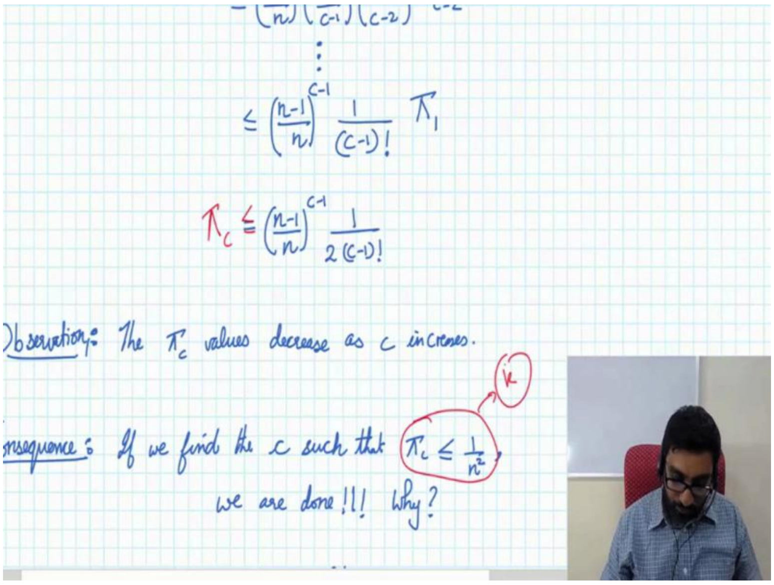
(Refer Slide Time: 10:59)



Now that we know pi 1. Let us look at how to compute pi c for some arbitrary c. So, again we are going to consider the cut between states 1 to c minus 1 and then state c to n. So, now let us look at the transition from left to right or rather from right to left. So, this is right to left. And, these pi n, pi n minus 1, up to pi c are the probabilities with which we will be in any one of these states n, n minus 1 and so on up to c.

And then, for each one of them with this probability 1 minus 1 over n times c minus 1 over n, we will be transitioning into the left side. And, we will be retrieving the whole left side as one virtual state. So, we do not know where we will be transitioning into. But, we will be transitioning into one of the states in the left side. And that is it should be equal to. And notice, while you are going from right or rather from left to right, you cannot transition from any state in the left to a state in the right. You can only go from c minus 1 to c. So, that is why here we only have pi c minus 1. And then, the probability that we will actually make that left to right transition is 1 over n.

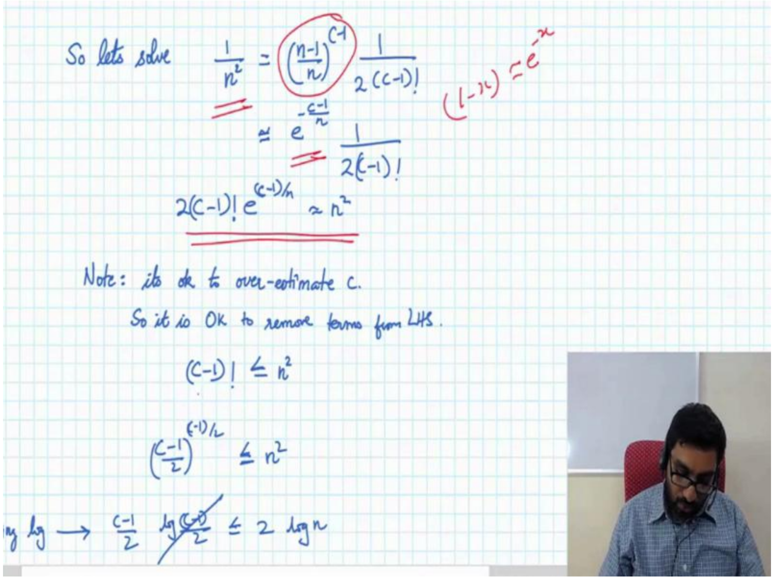
(Refer Slide Time: 12:49)



And here, if we work through this and make some simplifying assumptions. Say for example, all these probabilities are going to be greater than 0. And, so we simply ignore them and make this an inequality. And, we continue to solve this to get pi c to be less than or equal to n minus 1 over n raise to the c minus 1 times 1 over 2 times c minus 1 factorial.

Let us make a quick observation before we proceed. These pi c values, they decrease as c increases. And, what is the; And as a consequence, if we find the c value, if we find the c value such that pi c is at most 1 over n square, we are done; because we do not want a very large c value. We want as small as c value is possible. But, if we find a c value where pi c is 1 over n square, all subsequent values of pi c are only going to be smaller. And by the union bound, if you add up all of their probabilities it is not going to exceed 1 over n. And so, then the probability mass has to rest on all the states that precede at c, which is what we want. So, that is going to be the c. And that gives us the k. Basically, the c will get k.

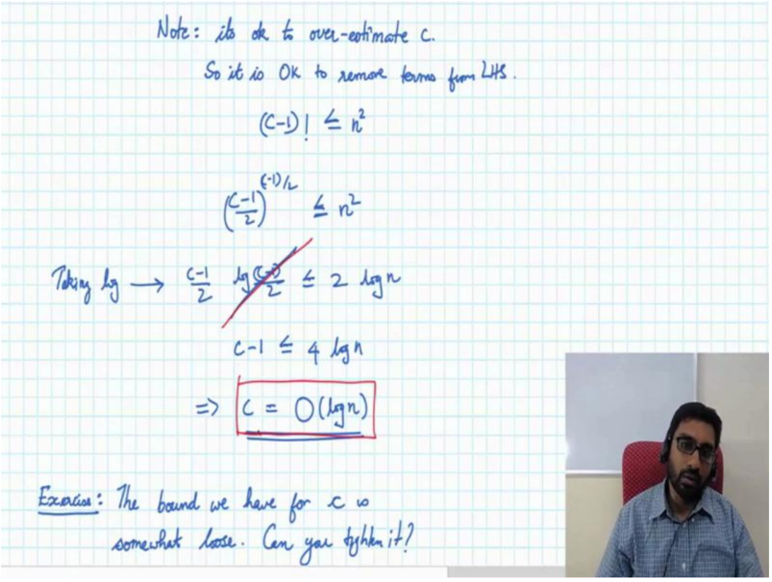
(Refer Slide Time: 14:38)



So with that being the case, let us try to solve for that c. So, we set; instead of pi c, we set it as one by n square. And, on the right hand side we have the expressions that we have over here. And, if we solve it and make some simplifying assumptions, we will eventually find the value of k. So, let us let us go through that. So this quantity, you are saying that 1 minus x is roughly equal to e to the minus x, when x is close to 0, we can use that to an approximate to e to the minus c minus 1 over n. And rearranging, we get this approximate equation. And, notice that it is okay to overestimate c because we just want; it is OK to be a little bit pessimistic.

So, what we are going to do is in order to get a way to find out c, we are going to ensure that we are going to throw away some terms. But, in that process we are only; we are going to ensure that we will only overestimate c. So, in the left hand side we are simply going to throw away some of the terms, so now c minus 1 factorial. So, we are just going to retain this one term; c minus 1 factorial is less than or equal to n square.

(Refer Slide Time: 16:23)



And, what is that equal to? Well, again we are going to do a little approximation on the left hand side, which will only will ensure that we will only overestimate the value of c. So, we are going to going to write c minus 1 factorial as c minus 1 by 2 raise to the power c minus 1 divided by 2. And that is at most n square. And now taking log, we get c minus 1 by 2 times log c minus 1 by 2 is at most 2 log n.

And, again we are going to throw away this term in order to get c minus 1 is at most 4 log n. Or, in other words c is O of log n. So, what is this? This, it says that the algorithm with high probability is going to output a celebrity that is within the top O of log n list of celebrities. So, that is the, that is the up short of this claim here. Of course, we have been fairly sloppy about the way we came up with this bound. So, this possibility that we can, we are bit more careful with the analysis to get a better bound. So, that will be left as an exercise for you.