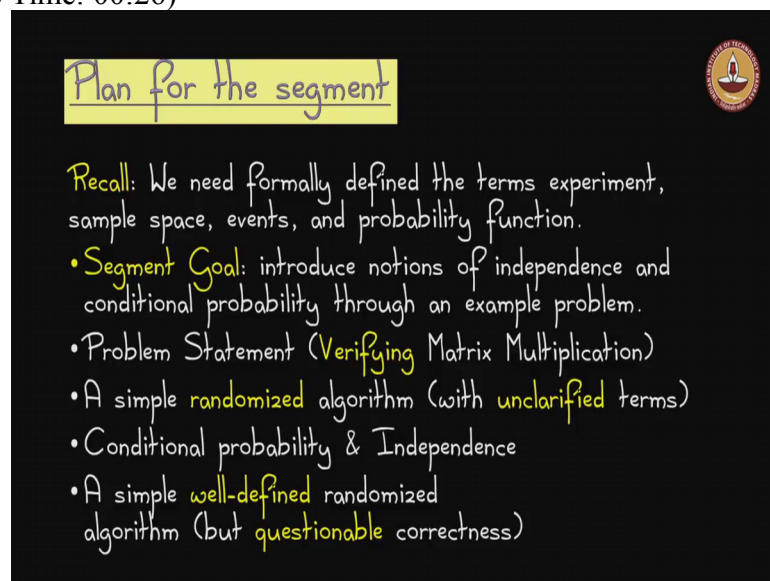



Probability & Computing
Prof. John Augustine
Department Of Computer Science and Engineering
Indian Institute of Technology, Madras

Module – 01
Introduction to Probability
Lecture - 03
Segment3: Verifying Matrix Multiplication (Statement, Algorithm, & Independence)

So, now we are going to start the third segment of the first module and in this segment well if you recall.
(Refer Slide Time: 00:26)





Plan for the segment

Recall: We need formally defined the terms experiment, sample space, events, and probability function.

- **Segment Goal:** introduce notions of independence and conditional probability through an example problem.
- **Problem Statement (Verifying Matrix Multiplication)**
- A simple **randomized** algorithm (with **unclarified** terms)
- **Conditional probability & Independence**
- A simple **well-defined** randomized algorithm (but **questionable** correctness)

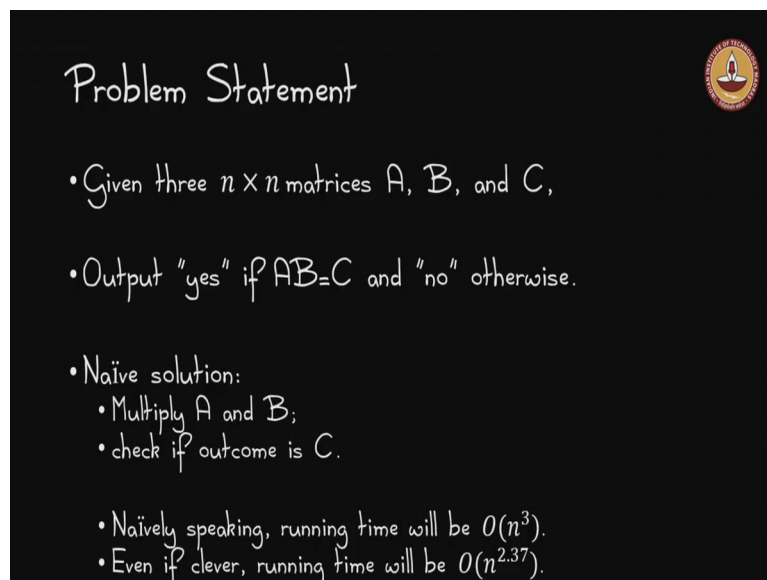
So, we now have a formal understanding of terms like sample space events probability function and so, on. So, the goal for this segment is to introduce the notion of independence, which is very fundamental in probability theory. And also the related notion of conditional probability and work through some examples.

And then we are going to look at and we are going to do this in the context of an algorithmic problem ok. So, it is going to be the problem of verifying matrix multiplication it is a and we are going to provide a very simple randomized algorithm.

And while we are doing that we are going to leave some undefined terms and that those terms the ones are going to lead to an understanding of conditional probability and independence.

We will try to clarify those terms and then finally, we end with a well-defined description of the randomized algorithm ok. We are going to still leave a few questions hanging at the end of this segment. For example, we will whether the algorithm is correct is going to be left out in this segment and we will address that in a subsequent segment.

(Refer Slide Time: 01:40)



Problem Statement

- Given three $n \times n$ matrices A , B , and C ,
- Output "yes" if $AB=C$ and "no" otherwise.
- Naïve solution:
 - Multiply A and B ;
 - check if outcome is C .
- Naïvely speaking, running time will be $O(n^3)$.
- Even if clever, running time will be $O(n^{2.37})$.

So, here is the problem statement, you are given 3 n cross n matrices A B and C . And you are asked to check if A times B equals C and you would not have you at the output yes if they are a B equals C and no otherwise ok. And of course, you all will know how to solve this very easily you just multiply A and B ok. There is only one downside to it this when you multiply you are going to take the normal naive multiplication is going to take theta of n cubed time ok.

And this is fine for small matrices, but nowadays the applications for matrix manipulation come from big data applications right, talking about very large matrices like 10 000 cross 10 000 or something like that.

And so, then an n cube time algorithm becomes prohibit prohibitive ok. So, you need to somehow speed this up there are some fancy algorithms that you can use you might have heard of strassens algorithm anybody. So, that brings it down to like 2 into the 2 point 8 9 or something like that.

And then even more fancy algorithms that only theoreticians think about will bring it down to n to the 2.37 I think there is been some improvement in the in this n to the 2.37. I think it was 2.3 8 and for like 20 years and then somebody came up with a very actually 2 people independently came up with brilliant ideas to improve it from n to the 2.3 8 to 2.3 7 or something like that.

So, yeah that is something you could do if you wanted to, but that is going to be very complicated algorithm, but our interest is to try and do this even more efficiently and take advantage of the fact that we can design algorithms that can toss coins ok.

(Refer Slide Time: 03:40)

Randomized algorithm

- Generate a column vector \vec{r} of n bits chosen uniformly and independently at random.
- Compute $AB^T \vec{r}$ and $C^T \vec{r}$, check if both equal.

What do these terms mean?

How can I trust the answer?

Note that $AB^T \vec{r}$ can be computed in $O(n^2)$. (How?)

So, here is how this randomized algorithm is going to work ok. We are going to generate a column vector and that is this vector r and we are going to do it in a very specific way we are going to choose there are n bits in this column vector each of those bits has to be uniformly at random.

So, another way of thinking about it is just flip a coin if it is heads make it a 1, if it is a tails make it a 0 it has to be a fair coin ok. And the outcome of and we also want each of these n bits to be independent of each other ok. This is again a term that we have not defined well, but it just means that these coins have to be done without influencing each other we will formally define the this term shortly ok.

So, you will get basically r_1, r_2, r_3 so, 1 of $2, r, n$ this is the column vector. And this we already have and the same column vector shows up here as 1 ok. And here is what we are going to do we are going to see whether $A B r$. So, matrix multiplication of a time A with B and then matrix 2 vector multiplication, which gives us $A B r$, whether we will check if that is going to be equal to $C C$ times r .

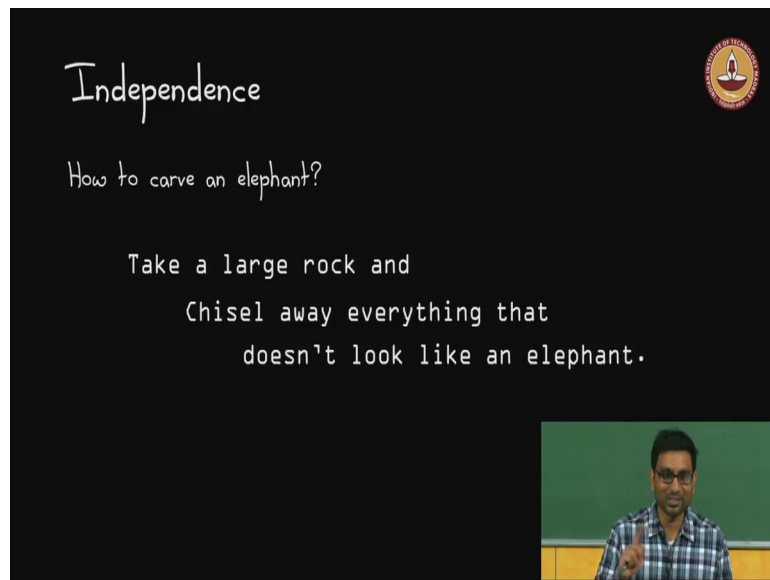
If these 2 are equal we are going to output saying $A B$ equal to C otherwise if they are if they turn out to be unequal we are going to output saying $a b$ is not equal to c ok. So, now, the nice thing that you should notice immediately is that $A B r$ can be computed in of n squared time why? Because you first perform this multiplication and that is going to be an n squared multiplication you will get a vector. And then again now you do you perform this multiplication ok.

This might be reminiscent of matrix chain multiplication that you might have studied in your algorithms right we are exploiting the same phenomenon here as well. And so, what now what we are going to get is an n squared time algorithm to compute the left hand side and of course, we have an n squared time to compute the right hand side as well. So, this algorithm it takes only off n squared time is, it very clear hopefully.

So, you just generate a random vector each element chosen uniformly and independently and random and you puff call that r you compute $A B r$ left hand side you compute $C r$ right hand side and check if they are equal. So, these are the terms that we need to worry about, but here is one other pesky question that should rankle you, why is this algorithm correct? Because we are asked to check whether $A B$ equal to r , but what we are doing is $A B r$ equal to $C r$ ok. So, that is something we need to worry about it will defer that question to A later side.

Now, I want to explain this notion of independence ok.

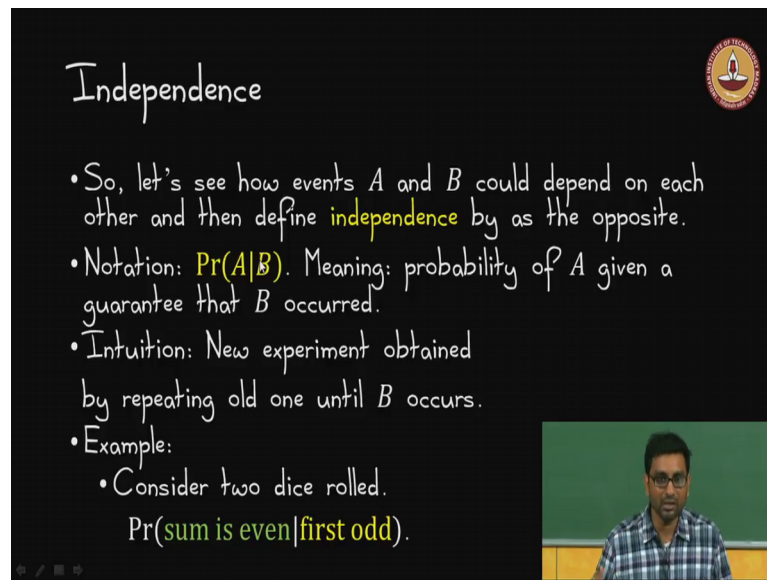
(Refer Slide Time: 07:07)



And in order to do that there is a very nice saying if you want to carve an elephant how do you do that? You take a large rock and chisel away everything that does not look like an elephant. And then what you are left with is going to be an elephant that is what we are going to do?

So, if you want to try and understand independence we are going to chisel away all notions of dependence and what we are left with is independence? We are going to take 2 events A B and we are going to see how they can depend on each other and define independence as the opposite ok. For that we will need A notation probability A given B ok.

(Refer Slide Time: 07:33)



The slide is titled "Independence" in a white, handwritten-style font. It contains several bullet points explaining the concept of conditional probability. The text is written in white and yellow on a black background. In the top right corner, there is a circular logo of the Indian Institute of Technology (IIT) Bombay. In the bottom right corner, there is a small video inset showing a man with glasses and a plaid shirt speaking.

Independence

- So, let's see how events A and B could depend on each other and then define **independence** by as the opposite.
- Notation: $\Pr(A|B)$. Meaning: probability of A given a guarantee that B occurred.
- Intuition: New experiment obtained by repeating old one until B occurs.
- Example:
 - Consider two dice rolled.
 - $\Pr(\text{sum is even} | \text{first odd})$.

So, this what does this mean it is the probability of an outcome A or an event A given A guarantee that outcome B occurred ok, given event B occurred ok. Another way to think of this is the following, you are running an experiment ok, you repeat the x and A and B are 2 events, you and remember what is the when is A and even you are talking about 2 subsets of the sample space.

You basically define A when you when you say A given B what you are doing is you are defining a new experiment ok. In this new experiment you are going to take the old experiment you are going to repeat it over and over and over again until B occurs the event B occurs. And when they even B occurs, then you check then you check I mean whatever outcome, you have that is the outcome of this new experiment ok. Does that make sense let me repeat what I am saying basically you start out with an experiment you are concerned about 2 events A and B ok.

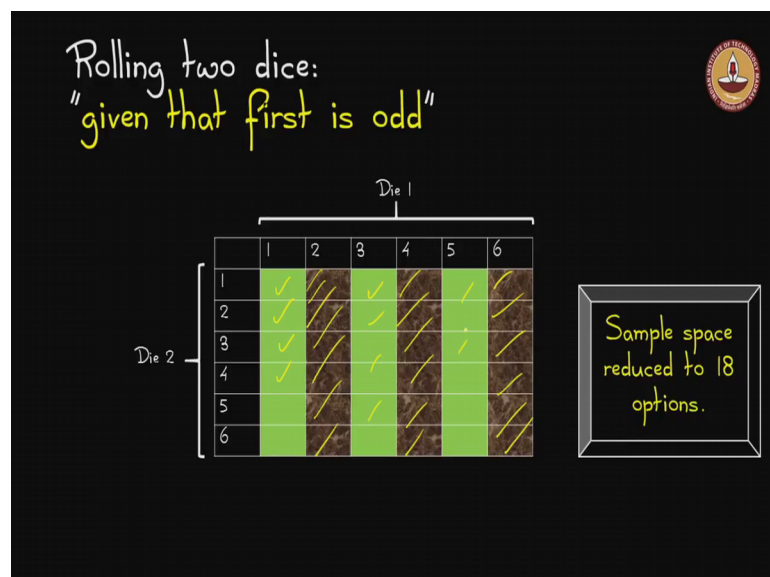
You want to understand the probability of A given B ok. So, the way to think about it at least one way to think about it is create a new experiment in, which you repeat this old experiment until B occurs. And will be A occurs whatever outcome you got in the last, when B occurred is the outcome of this new experiment ok. And in that experiment you want to understand the probability of A and that is this probability of A given B .

So, let us look at an example consider 2 dice that are rolled let us ask I mean. So, one way to exercise this understanding of this notation is we can ask, what is the probability that the sum is even given that the first outcome is odd ok.

So, let us say what you do is you roll 2 dice ok. And if the first outcome is even your condition is not satisfied B is called the condition right B is not satisfied ok.

So, that is not good enough you keep repeating the roll the 2 dice being rolled until the first dice is odd ok. The moment the first dice is odd, then you can ask is my some even and that is this notion of conditional probability.

(Refer Slide Time: 10:20)



So, let us work that out. So, now, you can roll 2 dice and you want to ask what is a the notion of given at, the first is odd another way of thinking about it is you look at the sample space when 2 dice are rolled ok. And then what you do is simply remove these outcomes because they are not relevant to this condition. If you condition that the first dice has to be an odd only these are the relevant outcomes ok. So, only the ones marked green.

So, now, ignore all the other outcomes and just worry about these outcomes and. So, in other words your sample space gets redefined ok. This is the out this is the sample space of this new experiment that I talked about ok.

(Refer Slide Time: 11:13)

Rolling two dice:
Sum is even given first is odd

		Die 1					
		1	2	3	4	5	6
Die 2	1	😊		😊		😊	
	2						
	3	😊		😊		😊	
	4						
	5	😊		😊		😊	
	6						

Sample space reduced to 18 options.

$$\frac{9}{18} = \frac{1}{2}$$

Now you can ask what is what is the probability that the sum is even given that the first is odd.

So, now you ignore all the those sample outcomes that are not relevant and then among those there are relevant mark green, you ask what are the ones that are what are the outcomes in which the sum is even and those are marked with a smiley face.

And now you can see that there are a total of 18 outcomes equally likely outcomes mark green out of which 9 have a smiley face. So, your probabilities are half.

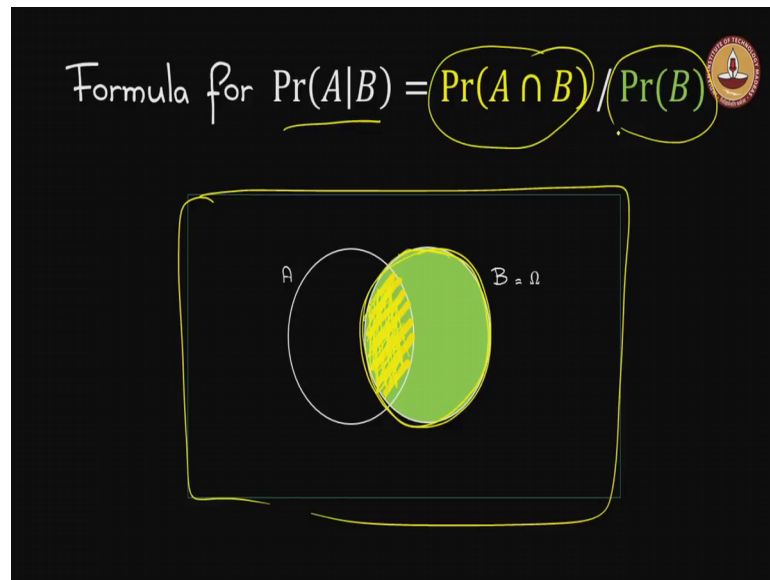
(Refer Slide Time: 11:51)

Formula for $\Pr(A|B) = \Pr(A \cap B)$?

Almost. We need to reduce Ω to B

So, now let us try to arrive at a formula for this probability of A given B let us first do try something ok. So, let us first try probability of A intersection B ok. This is this is not quite hitting the mark, because there is this other aspect to this conditional probability right. We need to reduce the sample space somehow ok.

(Refer Slide Time: 12:26)

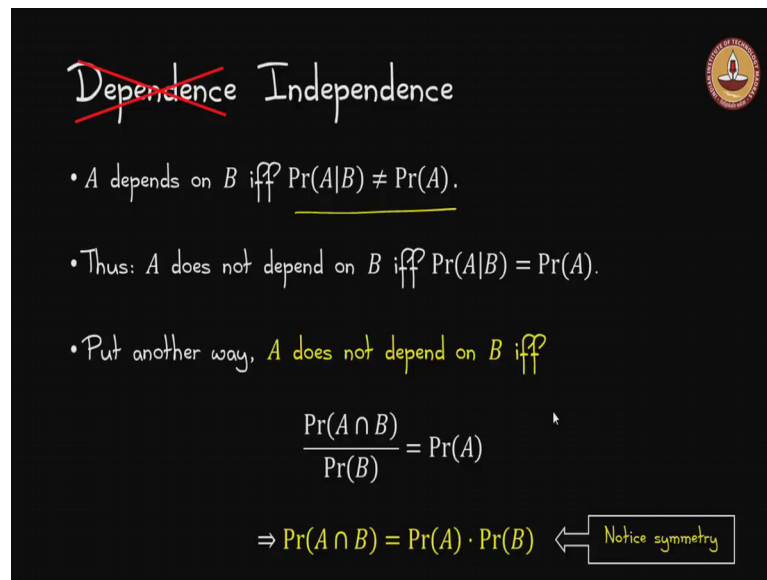


So, what we are going to have to do is probably so, basically now you have to throw away all the sample space outside of B. So, when you talk about probability of A given B what you have to do is consider this as your sample space and out of that you consider what is the probability that A occurred? Given that B occurred ok.

So, your sample space is B and out of that sample space what is the probability that your event of interest A occurred ok. And how do you achieve that you this probability of A intersection B gives you this portion, but then this is with respect to the overall sample space.

So, now you have to restrict it to just B you achieve that by in some sense normalizing over B. So, that is basically dividing by B ok. So, this is your formula for computing probability of A given B and I could have just written the formula and walked away, but I would like you to think about how these formulas come about ok.

(Refer Slide Time: 13:47)



~~Dependence~~ Independence

- A depends on B iff $\Pr(A|B) \neq \Pr(A)$.
- Thus: A does not depend on B iff $\Pr(A|B) = \Pr(A)$.
- Put another way, A does not depend on B iff

$$\frac{\Pr(A \cap B)}{\Pr(B)} = \Pr(A)$$

$\Rightarrow \Pr(A \cap B) = \Pr(A) \cdot \Pr(B)$ ← Notice symmetry

So, now let us talk about dependence remember we are going to have to we have to understand dependence to chisel it away. So, that we left with independence.

So, it is easy to also think about dependence right if when can we say that A depends on B , if somehow when B occurs some the probability of A changes. Then you know that A depends on B the fact that B occurred somehow change the out the probability of A , then you know that is dependent on B . So, that is what we are going to write in this fashion.

So, we so, simple now we are ready to chisel it away it. A does not depend on B oops if probability of A given B is equal to the probability of A now the thing is. So, basically A does not depend on B now we have A formula for probability of A given B right, just applying that formula we get property of A intersection A does not depend on B , if and only if probability of A intersection B divided by probability of B equals probably.

So, now we will come the classic formula or definition of independence this you might recall is the way diff independence is often defined and even A is independent of B if probability of A intersection B equals to the product of the 2 individual probabilities probability of A times probability of B ok. So, this is you might have seen this as the very definition, but hopefully the intuition is clear as to how your arriving at this definition.

And here you should notice something we are talking about A depending on B, but look at the condition for independence can you interchange A and B and still have the notion. So, that is there is this beauty here there is A symmetry here if A is independent of B, then what can you also say B is also independent of A.

(Refer Slide Time: 15:51)

~~Dependence~~ Independence

Recall: $\Pr(A \cap B) = \Pr(A) \cdot \Pr(B)$

• What about B?

$$\frac{\Pr(A \cap B)}{\Pr(A)} = \Pr(B|A) = \Pr(B)$$

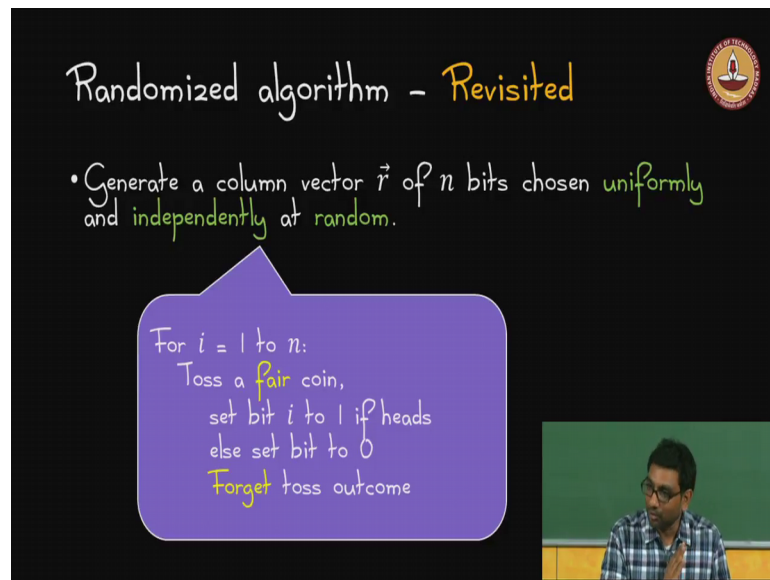
Summary: A & B independent iff

- $\Pr(A|B) = \Pr(A)$ or
- $\Pr(B|A) = \Pr(B)$ or
- $\Pr(A \cap B) = \Pr(A) \cdot \Pr(B)$

Thus independence is symmetric: B also does not depend on A iff A does not depend on B.

So, you can actually work out the exact same notion for B and you will get that thus independence is A is symmetric and B also does not depend on A, if and only if A does not depend on B. So, to summarize we A and B are independent if probability of A given B equal to A property of B given A is equal to B and probability of A intersection B equal to property of A times B all these are equivalent statements any questions on the notion of independence?

(Refer Slide Time: 16:33)



Randomized algorithm - Revisited

- Generate a column vector \vec{r} of n bits chosen uniformly and independently at random.

For $i = 1$ to n :
Toss a fair coin,
set bit i to 1 if heads
else set bit to 0
Forget toss outcome

The slide features a logo in the top right corner and a small video inset in the bottom right corner showing a man speaking.

So, now we can revisit this algorithm. So, would we have to generate A column vector r of n bits chosen uniformly and independently at random. And how do we do that we know the notion of independence how do we actually do that.

So, now, what you do is for each random bit we have to do this for i equal to one to n toss A fair coin set the bit i to 1 if heads else bit to 0 and completely forget the outcome of the toss, because the next one has to be completely independent of the previous pointer.


So, comes and you repeat this n times you will get vector r that you need.

(Refer Slide Time: 17:19)

Randomized algorithm - Revisited

Claim: Each of the 2^n binary column vectors is equally likely

Consider any fixed binary column vector $(r_1, r_2, \dots, r_n)^T$.
Since each bit generated independently, we generate this fixed vector with probability $\left(\frac{1}{2} \times \frac{1}{2} \times \dots \times \frac{1}{2}\right) = 2^{-n}$.



Now, there is an important claim here each of the 2^n binary column vectors now if you have a binary vector of n bits. How many outcomes I mean what are the total number of possible such binary vectors is 2^n right. Each of the 2^n binary column vector is equally likely, we want to be able to argue that each one of them is equally likely right.

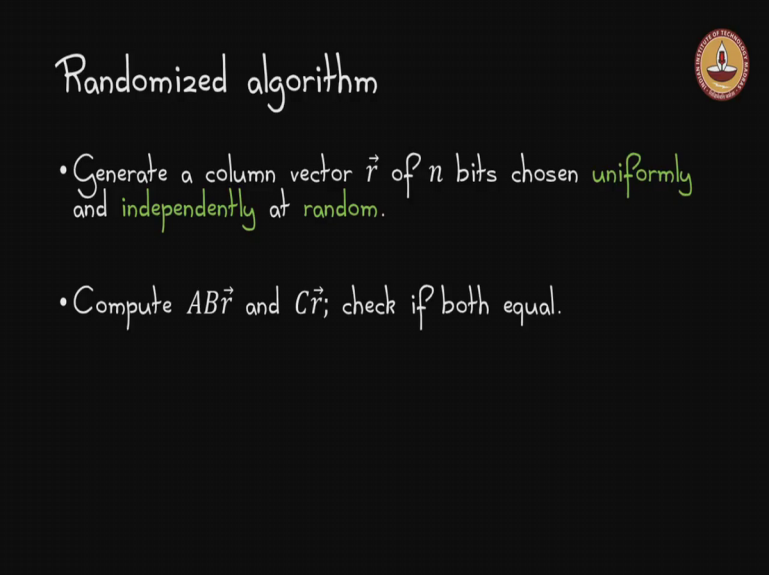
So, what we know is each individual bit is equally likely to be 0 or 1. How does that imply that the entire n bit binary vector is drawn from a sample space of 2^n binary strings each equally likely that is the question ok. So, now, how we are going to do it remember now our sample space is 2^n binary vectors ok, pick 1 any arbitrary 1 we call it r_1, r_2 and so on up to r_n ok.

So, a vector sum using the column vectors and using transpose over them, remember each bit was generated the way you generated you generated completely independent of each other right. So, and if they were independent you recall that they are probably individual the probability of r_1 being a certain value r_2 being a certain value and so on those individual probabilities can be multiplied, because that is the formula for independence we showed it for 2 events, but it extends to any events as well ok.

So, then you take one by for each r_1 the probability that you get the appropriate r_1 is half appropriate r_2 is half and so on. So, you multiply half of it n times you get 2^{-n} . So, what have we established if you pick any one of the 2^n

power n binary vectors you the probability that you get that particular binary vector is one by 2 raised to the n that is equally likely.

(Refer Slide Time: 19:24)



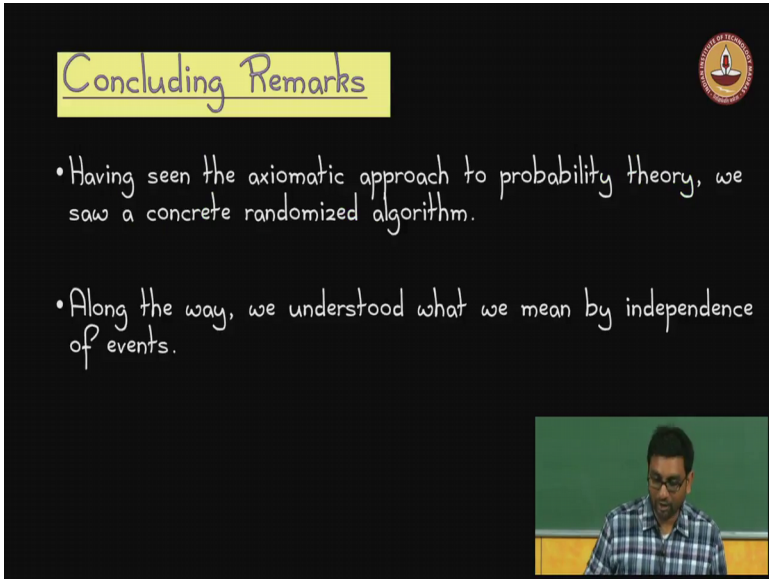
Randomized algorithm

- Generate a column vector \vec{r} of n bits chosen uniformly and independently at random.
- Compute $AB\vec{r}$ and $C\vec{r}$; check if both equal.

The slide features a black background with white and green text. A small circular logo is in the top right corner. The title 'Randomized algorithm' is written in a white, handwritten-style font. The two bullet points are also in a white, handwritten-style font, with the words 'uniformly' and 'independently' highlighted in green.

So, now hopefully if we revisit the randomized algorithm all the terms are clear to us.

(Refer Slide Time: 19:35)



Concluding Remarks

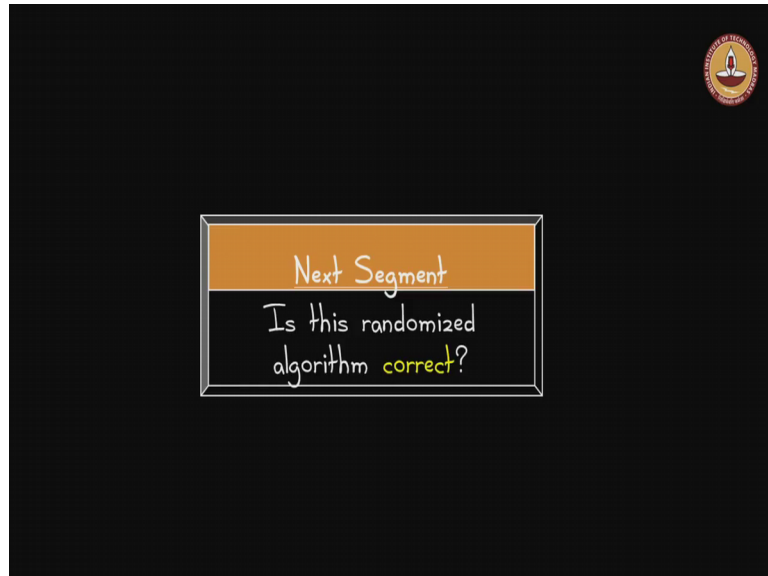
- Having seen the axiomatic approach to probability theory, we saw a concrete randomized algorithm.
- Along the way, we understood what we mean by independence of events.

The slide features a black background with white text. A yellow highlight box is behind the title 'Concluding Remarks'. A small circular logo is in the top right corner. The two bullet points are in a white, handwritten-style font. In the bottom right corner, there is a small video inset showing a man in a plaid shirt speaking in front of a green chalkboard.

We know exactly what we mean by uniformly and independently and then all the algorithm is clear to us we have understood, but either way along the way we have

understood the notion of independence of events conditional probability, we still do not know whether it is correct.

(Refer Slide Time: 19:42)



We know the running time. So, we know this k is guaranteed to run in $O(n^2)$ time we do not know the correctness and we will see that in the next segment ok.

Thank you.