

**Probability for Computer Science**  
**Prof. Nitin Saxena**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology - Kanpur**

**Module - 4**  
**Lecture - 13**  
**Equality Checking. Poisson Distribution.**

Last time we were doing this nice example of continuous random variable, which is called Buffon's needle problem. So, on infinite floor, there are parallel lines, let us say 1 centimetre apart, and you drop a needle of some length 1 centimetres. So, the needle falls in a random orientation and the random variable of interest is number of intersections with the lines on the floor. So, what is the expectation of this random variable?

So, we argued that this is nothing but an application of linearity of expectation. So, you can actually break the needle into smaller parts, as small as you like, and then independently calculate the expectation for that part. So, then, the needle actually stops being a needle; you can think of it as any figure; and we agreed that we will look at it as a circle, because circle has the advantage that wherever it falls, the number of intersections does not change.

**(Refer Slide Time: 01:27)**

$\Rightarrow$  Pick a shape that gives the same #intersections wherever it drops!

$\triangleright$  Circle of diameter = 1cm always gives 2 intersections!

$\triangleright$   $E[X] = 2$ , where the "length" =  $2\pi \cdot \frac{1}{2} = \pi$  cm.

$\triangleright$   $E[X] = \frac{l}{\pi} \times 2.$   $\square$

- Let's see an example that is very useful in computer science:

So, last time we saw that, if we drop a circle of diameter 1; so, circle of diameter equal to the distance between the lines, which is 1 centimetre, always gives 2 intersections. So, if it falls directly in the middle of the 2 lines, then it gives you these as tangent points; if it falls somewhere else, then you get an intersection with a single line. So, it is always 2. And what

this means is, expectation of this circle intersection, let us call it  $X$  prime, this is fixed, it is just 2; where the length is equal to the circumference, so, it is  $2\pi r$ ;  $r$  is half; so, it is  $\pi$ .

So, for  $\pi$  centimetre long needle, you expect intersections to be always 2, which means that expectation of that random variable, original one, for 1 centimetre length, this is 1 by  $\pi$  times 2. So, that finishes the problem. So, for a needle of length 1 centimetre, you expect 2 by  $\pi$  as the expected number of intersections. This is an amazing application of linearity of expectation.

I wanted to do one more random variable which is famous, but before that, let us continue with another example. So, let us see an example that is very useful in computer science, and this will be equality checking protocol.

(Refer Slide Time: 04:23)

Equality Checking Protocol

Alice  $\longleftrightarrow$  Bob

Files: A B

- Alice (Bob) has a file A (resp. B). Each file is  $n$ -bits long.
- Design a communication protocol to test  $A \stackrel{?}{=} B$ , by sending as few bits as possible.

Brute-force: Alice sends A to Bob. #bits =  $n$ . Better?

Deterministic protocol not possible?

- We give a probabilistic one, using number theory!

Protocol: 1) Turn A into number  $N_A := \sum_{0 \leq i < n} a_i \cdot 2^i$ .

2) Pick a random prime  $p \in [t]$ .

So, here the setting is, there are 2 players, Alice and Bob. Physically they are separated, so, they have some channel of communication and they have files. So, Alice has a file A and Bob has a file B, and there is a channel of communication between them, both ways. So, write that down; so, Alice has each file is  $n$  bits long; so, generally  $n$  is very large. Usually, files are thousands and millions of bits long, so,  $n$  is some large number.

So, Alice would not want to send this, the whole file through the channel. So, what you have to do is, design a communication protocol to test whether A and B are the same files, by sending as few bits as possible. So, few bits is the parameter to optimise in this question. The

number of bits which is sent, that should be as few as possible. So, obviously, Brute-force is just, Alice sends the whole file; but then, number of bits will be  $n$ .

So, you want to do it better than this. So, can Alice manage equality checking in bits much smaller than  $n$ ? Actually, even  $n - 1$  is not clear at this point. So, what is clear is, deterministically, you cannot improve this. So, it seems very hard to create a deterministic, to design a deterministic protocol, which requires much less than  $n$  bits to be transferred, and still  $A$  and  $B$  equality gets tested. So, let us give a probabilistic protocol.

So, we give a probabilistic one using number theory, which seems suspicious. So, how can you use number theory in this problem? This is just a question about bits, just random files or arbitrary files. So, the idea will be to actually see file  $A$  as a number. So, let us design a protocol based on that. So, turn  $A$  into number  $N_A$ , just the number that whose bit representation binary representation this is; so, which is; a  $i$  is the  $i$ th bit;  $2$  raised to  $i$ ;  $i$  is  $0$  to  $n - 1$ , that is the number which the file  $A$  represents.

Second, pick a random prime  $p$  in some range; I am not telling you the range; let us just use this placeholder  $t$ . So,  $1$  to  $t$ , pick a prime number. So, the prime can be  $2$ ; it can be  $3$ ; it can be  $5$ ; it can be  $7$  and so on. So, immediately the question arises that, how many primes are there in this interval  $1$  to  $t$ . So, that we will have to handle later. Let us continue with the protocol.

(Refer Slide Time: 10:13)

3) Compute residue  $R_A := N_A \bmod p$  [ $\Rightarrow \lg t$ -bits]

4) Send  $(R_A, p)$  to Bob.

5) Bob checks  $R_A \stackrel{?}{=} R_B$ . [Output Yes iff  $R_A = R_B$ ]

Qn: What's min  $t$ , to get a "good" success probability?

Analysis:

$\pi(t) := \# \text{primes in } [t] \approx \frac{t}{\lg t}$  [Why?]

(Prime number thm.)

$\triangleright P(R_A = R_B \mid A = B) = 1.$

$\triangleright A \neq B \Rightarrow (N_A - N_B) \neq 0$  has at most  $\lg |N_A - N_B| < n$  prime factors.

So, the main step in the protocol is computing a residue,  $N A \bmod p$ . So, this is,  $p$  was at most  $t$ , so, it is  $\log t$  bits. So, the residue is only  $\log t$  bits. So, if you pick  $t$  to be small, then these are few bits. So, you have basically mapped the bit file,  $n$  bit file to few bits, just  $\log t$  bits. And then, you send this residue  $R A$  and the prime  $p$  to Bob. So, this again is only  $2 \log t$  bits; so, it is very few amount of bits, small amount of bits which have been sent over the channel.

And then, Bob checks whether  $R A$  is equal to  $R B$ . Bob can do that because Bob has the prime number  $p$ ; and he obviously has the file  $B$ . So, he computes this integer  $R B$ ; reduces this; find the residue mod  $p$ ; and checks whether the residue is the same. So, obviously, you output yes, if and only if  $R A$  is  $R B$ . That is the output. If  $R A$  and  $R B$  are different, then clearly, the files were different; but if these residues are the same, can you say with confidence that the files were the same?

Because, there might not be enough information in  $R A$ ,  $R B$ . So, they may be the same; files may be different. So, that error probability, we have to calculate. So, what is the minimum  $t$  that will work to get a good success probability? So, that is our goal. So, to really fix the algorithm, we have to do an analysis. Find  $t$ , fix  $t$  and then also say what is the confidence that we have.

Given  $R A$  is equal to  $R B$ , with how much confidence can we say  $A$  equal to  $B$ ? So, let us do the analysis. So, first thing that you have to learn is, how many primes are there till  $t$ ? So, this is number of primes in the set  $1$  to  $t$ , and that is around  $t$  over  $\log t$ . So, that is a deep question. I will not resolve it here. In case you do not know, you look at the density of primes; there is prime number theorem; look at that.

So, that says, number of primes will be  $t$  over  $\log t$ , which is; if you look at the density, it is  $1$  over  $\log t$ . So, as  $t$  grows, the density really falls, but still its number of primes is actually not that bad; it is a good density. And definitely, as this says, there are infinitely many primes. There is no dearth of primes. So, other property is that probability that  $R A = R B$  in case  $A = B$ ; the files are equal; that is certainly  $1$ .

If the files are the same, the numbers are the same, so, the residues are the same; they cannot be different. Now, let us look at the case when  $A$  and  $B$  are different. The files are different,

then what is happening? In that case, we want to evaluate the chance that residues are different. If the residues are the same, that is an error. So, if files are different, then the numbers are different.

The number cannot be the same, because it was just the binary representation. And it has, this number has at most, how many prime factors? So, that will be log of this number's value. This is a non-zero number. And if you look at the; its absolute value, take the log; the number of primes cannot be more than this, prime factors; because, if it has  $n$  prime factors, then each prime factor is obviously, at least 2.

So, the prime factors are 2, 3, 5, 7. All of them are at least 2; so, you will get 2 raised to  $n$ , which is a number that is bigger than this, the difference. So, small number, I mean this number has magnitude at less than 2 raised to  $n$ ; so, it cannot have more than  $n$  prime factors. So, there are few prime factors. So, what does it say about the probability?

**(Refer Slide Time: 17:14)**

$$\Rightarrow P(R_A = R_B \mid A \neq B) = \frac{\# \text{prime factors}(N_A - N_B)}{\# \text{primes in } [t]}$$

$$< n / \pi(t) \leq \frac{n \log t}{t} \quad [\text{Fix } t := 4n^2 \log n.]$$

$$= n \cdot \frac{\log(4n^2) + \log n}{4n^2 \log n} < n \cdot \frac{3 \log n}{4n^2 \log n} < \frac{1}{n}.$$

Thm: The protocol transmits  $2 \log t = O(\log n)$  bits & succeeds with probability  $> (1 - \frac{1}{n})$ .

↳ Note that  $(\log n)$ -bits needed to even index a bit in file A. So, the protocol is amazingly efficient!

So, this says that probability that  $R_A$ , residue of A and residue of B are the same, given that the numbers were different, the files were different. This is equal to the number of prime factors of  $N_A$  minus  $N_B$  divided by the number of primes in this universe 1 to  $t$ . So, this is less than  $n$ . And number of primes, we are calling  $\pi(t)$ . So,  $\pi(t)$  we said is around  $t$  over  $\log t$ . So, let us put that. So, that is the error probability,  $n \log t$  by  $t$ .

So, let us take  $n$  to be; fix  $t$  to be  $4 n^2 \log n$ , because, if you fix  $t$  to be this, then the error probability becomes  $1$  by  $n$ . So, this is  $n$  times  $\log t$ .  $\log t$  is  $\log$  of  $4 n^2 \log n$  plus  $\log$

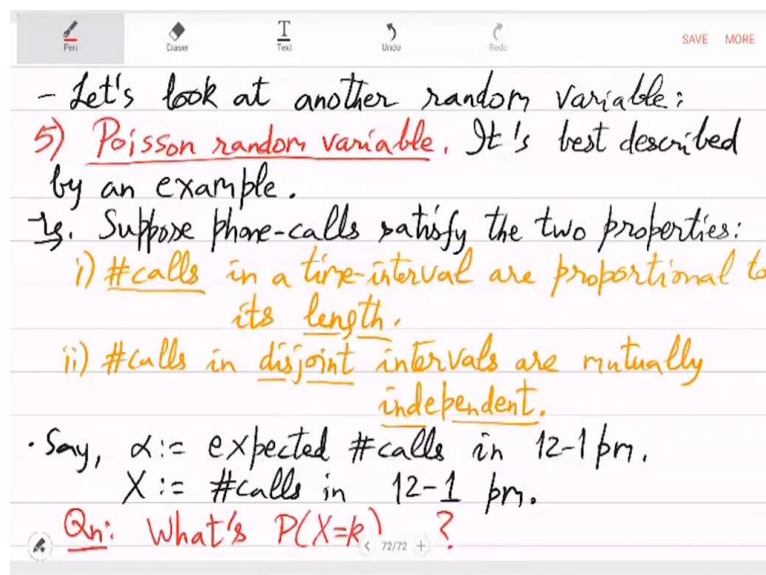
$\log n$  divided by  $4n \log n$ . So, this is less than  $n$  times; so, the dominating term is like  $2 \log n$ ; slightly more, so, let us say  $3 \log n$  divided by  $4n \log n$ , which is less than  $1/n$ . So, this becomes the error probability  $1/n$ .

So, what we have shown is that the protocol transmits  $\log t$  or  $2 \log t$ , which is;  $\log t$  is what?  $\log n$ , right? So, it is order  $\log n$  many bits and succeeds with probability greater than  $1 - 1/n$ . So, if  $n$  is thousands and millions of bits, then this success probability is almost 100%. So, by sending only  $\log n$  many bits, you are able to get success almost 100%. So, this is an amazing protocol; especially note the optimality of this.

This is really an optimal algorithm, because; note that  $\log n$  bits are needed to even index a bit in file A. So, file A has  $n$  bits. So, if you want to just locate a bit, that location already takes  $\log n$  bits, just to address, just to point a bit. And around that much space, you are able to check the equality of A and B. So, this protocol is amazingly efficient. So, that is a very good example of the use of probability.

And with this, let me finish up with the last famous or important random variable. So, what you have seen till now in the discrete setting were 4 random variables, Bernoulli, binomial, geometric and negative binomial. The last you will see is Poisson.

(Refer Slide Time: 22:31)



- Let's look at another random variable:  
5) Poisson random variable. It's best described by an example.  
- Eg. Suppose phone-calls satisfy the two properties:  
i) #calls in a time-interval are proportional to its length.  
ii) #calls in disjoint intervals are mutually independent.  
- Say,  $\alpha :=$  expected #calls in 12-1 pm.  
 $X :=$  #calls in 12-1 pm.  
Qn: What's  $P(X=k)$ ?

So, this is number 5 in discrete random variables, is called Poisson random variable. So, what is this? So, it is best described by an example, because, if I give you the formula, it will not make any sense. So, let me describe this by an example. It is a very natural setting. So,

suppose, look at cell phone calls. So, suppose phone calls satisfy the 2 properties as follows. So, one is that number of calls in a time interval are proportional to the length of the interval.

This is completely natural, maybe not in the night, but during the day, let us say 12 to 1 p.m. or let us say 12 to 4 p.m. It is natural that, if you look at 12 to 1, and then you look at 12 to 2, then the number of phone calls should double. So, number of calls proportional to length. And second is, number of calls in disjoint intervals are independent. So, what this is saying is that, the number of calls you see 12 to 1 p.m. and the ones you see 2 to 3 p.m., they are independent; which again seems to be reasonable, because these are different times; so, why should there be a correlation between the 2?

I mean, if you look at the whole country, then, there should not be any correlation. Maybe in an office, there is some correlation, but across a big sample space, geographically distributed, there should not be any correlation. So, these are the 2 things you can assume about phone calls. And now, so, say  $\alpha$  is the expected number of calls in the time duration or interval 12 to 1 p.m.

And  $X$  is the number of calls; so, it is a random variable; it is the number of calls in 12 to 12:15 p.m.; or let me use the same time. So, question is, what is the probability that  $X$  is equal to  $k$ ? So,  $X$  is a random variable; its expected value is  $\alpha$ ; but what is the probability that it is  $k$ ? And  $k$  is some number, 0, 1, 2, 3. So, what is this function? We are interested in this function.

So, this seems very tricky. I mean, you can calculate the expected number of calls by taking averages every day; but now, you want to make a prediction for the, let us say, next week; next week, some day, 12 to 1 p.m., what is the probability that number of calls is  $k$ . So, assuming these 2 properties, this can be worked out beautifully. So, let us see; this should surprise you.

**(Refer Slide Time: 28:00)**

• Because of the continuous nature of calls, we divide the interval into  $n$  discrete parts:

$P(\text{call in one part}) = \alpha/n =: p.$

$$\triangleright P(X=k) = \binom{n}{k} \cdot p^k \cdot (1-p)^{n-k} = \frac{n(n-1)\dots(n-k+1)}{k!} \cdot \left(\frac{\alpha}{n}\right)^k \cdot \left(1-\frac{\alpha}{n}\right)^{n-k}$$

$$= (1-\frac{1}{n}) \cdot (1-\frac{k-1}{n}) \cdot \frac{\alpha^k}{k!} \cdot (1-\frac{\alpha}{n})^n \cdot (1-\frac{\alpha}{n})^{-k}$$

$$\lim_{n \rightarrow \infty} = 1 \cdot \frac{\alpha^k}{k!} \cdot e^{-\alpha} \cdot 1 = \frac{\alpha^k}{k!} e^{-\alpha}.$$

$$\triangleright \sum_{k \geq 0} P(X=k) = \sum_{k \geq 0} \left( e^{-\alpha} \frac{\alpha^k}{k!} \right) = e^{-\alpha} \cdot e^{\alpha} = 1.$$

$$\triangleright E[X] =$$

So, those 2 properties make the calls continuous in nature. So, because of the continuous nature of calls, we divide the interval into  $n$  discrete parts. So, think of  $n$  as something very large. And then, in 1 part, only 1 phone call can come. What is the probability of that phone call actually being made? So, that probability, by the continuous nature or by these 2 properties, the phone call in a part is  $\alpha/n$ .

$\alpha$  was there in that 1 hour; then you divide that 1 hour into  $n$  parts; in each part, a phone call actually being made is  $\alpha/n$ . So, if it in 1 hour, there were 10 phone calls and you divide this into 20 parts, then in 1 part, the probability of a call being made is half. That is why, in 20, summing over 20, you will get 10. It cannot be more; it cannot be less; so, this seems intuitively correct. And that is your probability  $P$ .

So, let us now do the calculation of what we want, which is  $X$  equal to  $k$ . So, that is, in  $n$ , you want  $k$  parts to be active. So, that is  $n$  choose  $k$ , many ways. In  $k$ , call happens; in  $n - k$ , does not happen. So, in  $k$ , you want  $p$  raised to  $k$  probability, or you have  $p$  raised to  $k$  probability; otherwise, it is  $1 - p$  to the  $n - k$  probability. So, what is that? So, that is  $n \dots n - k + 1$  by  $k$  factorial.

$p$  is  $\alpha/n$  to the  $k$  and  $1 - \alpha/n$  to the  $n - k$ . Let us rearrange. So, you will get  $1 - 1/n \dots 1 - k/n$  and  $\alpha^k/k!$  and these two things. So, now, obviously, there is no good number  $n$ ;  $n$  can be taken to be anything. You can take it to be 1 second; but then, it is also, if the population is large, then it is possible that actually in half a second there is a call.



So, you can take it to be half a second or you can take it to be 1 millisecond. So, essentially,  $n$  is infinity,  $n$  is tending to infinity. So, in that case, what is this expression? So, limit of this as  $n$  tends to infinity is, the first whole part is 1;  $\alpha$  to the  $k$  by  $k$  factorial, it will not change.  $1 - \alpha$  by  $n$  to the  $n$ ; so, recall your calculus; so, this is  $e$  to the minus  $\alpha$  in the limit. And the the last thing is  $1 - 0$ ; so, it is just 1.

So, what you get is  $\alpha$  to the  $k$  by  $k$  factorial,  $e$  raised to minus  $\alpha$ . That is the answer. So, the probability that  $k$  calls will be made is given by this. So, that is a weird expression. Let us try to see some properties. First property of this expression is, if you look at probability going over all the  $k$ 's, then what do you get? So,  $e$  raised to minus  $\alpha$ ,  $\alpha$  to the  $k$  by  $k$  factorial.

So, this expression is actually  $e$  raised to minus  $\alpha$  times  $e$  raised to  $\alpha$ . The definition of  $e$  raised to  $\alpha$  is this expression. So, you get 1. And the second is, you can also calculate the expectation of  $X$ . So, this we will do next.