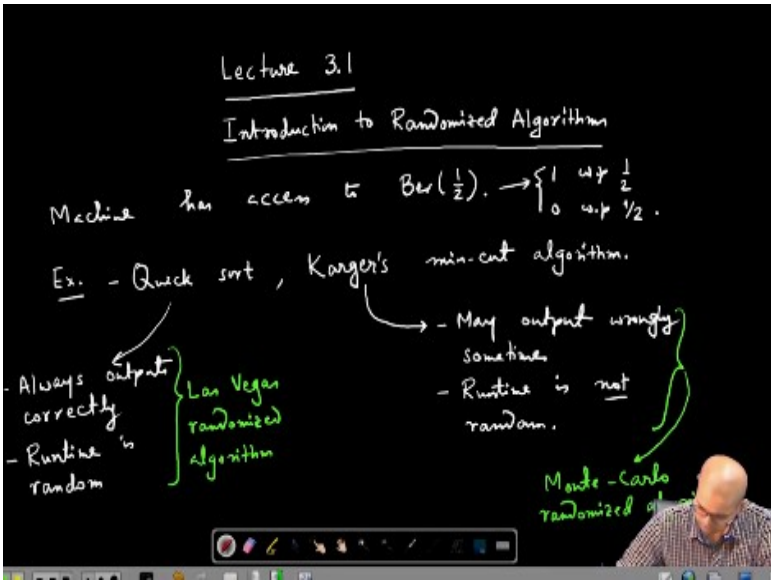
Selected Topics in Algorithm
Prof. Palash Dey
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture No # 11
Module No # 03
Introduction to Randomized Algorithm

Welcome so in the last class we have seen the analysis of Edmond's Blossom algorithm for finding maximum matching. Now from this class we will start looking at some randomized algorithms.

(Refer Slide Time: 00:43)



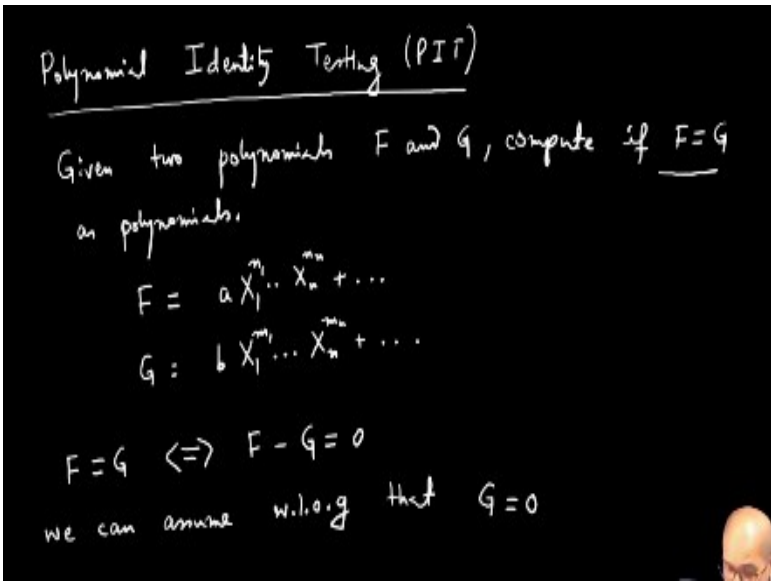
So lecture 3.1 so suppose our machine has access to Bernoulli random variable a variable which Bernoulli half which is one with probability half takes value 1 with probability half and is 0 with probability half. Means access to fair coin it can toss fair coin and we have seen that randomization often simplifies the algorithm. For example we have seen we all know randomized quick sort then Karger's min-cut algorithm.

Now if you observe closely there are fundamental difference between the randomized quick sort and Karger's min-cut algorithm. What randomized quick sort does is? It always output correctly but runtime is a random variable runtime can vary, runtime is random. On the other hand

Karger's min-cut algorithm this is; may output wrongly sometimes but runtime is fixed runtime is not random runtime is not random.

The first type of algorithms they are called Las Vegas randomized algorithms and the second one this Karger's type of algorithm are called Monte Carlo randomized algorithm good. So now we will see some more Monte Carlo type of randomized algorithms.

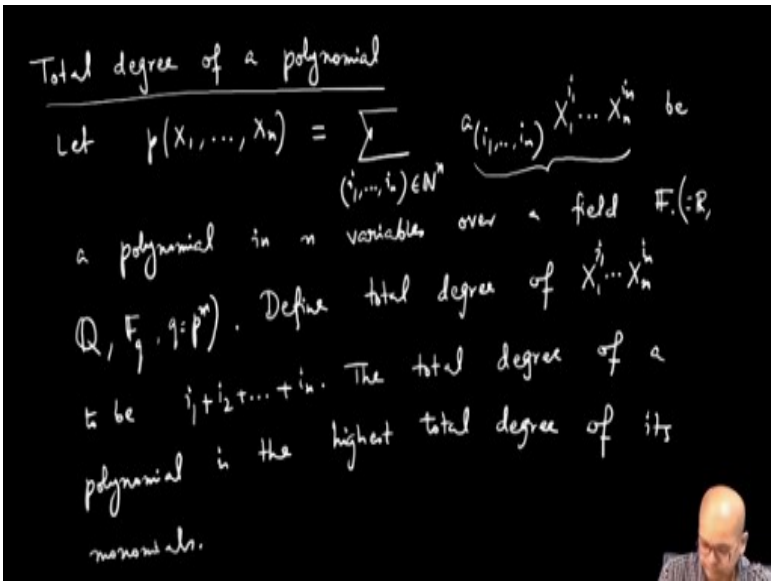
(Refer Slide Time: 05:31)



Our first example is polynomial identity testing pit. So what is the problem? So given 2 polynomials F and G check or compute if F = G, as polynomials. So this whether so F is a polynomial, so F is a multivariate polynomial so it has some coefficient a times $X_1^{n_1} \ldots X_n^{n_n}$ and so on this is how it looks like. And G also like that b times $X_1^{m_1} \ldots X_n^{m_n}$ and so on so these 2 polynomials as polynomials are identical this is the question.

First observe that checking whether F = G is equivalent to checking is F and only if F - G is 0. So in the polynomial identity testing problem we can assume without loss of generality that G = 0 I am simply given one polynomial F and I need to test whether F = 0 or not whether it is a 0 polynomial or not.

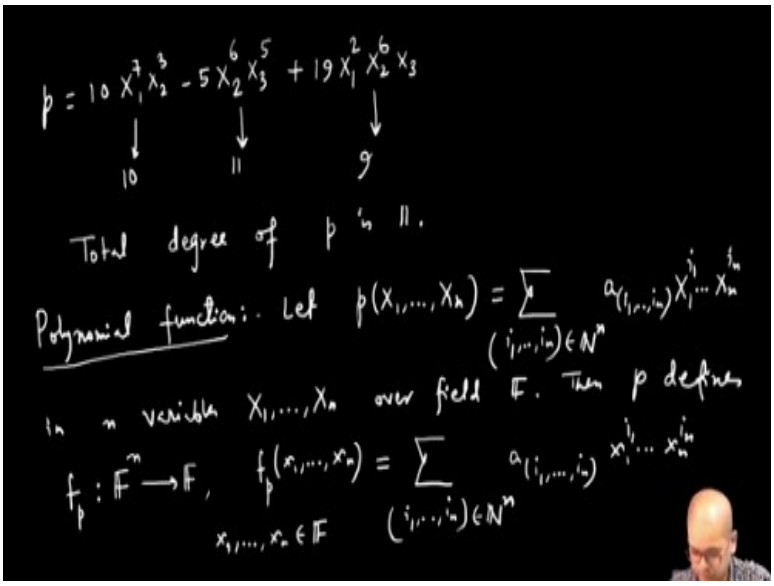
(Refer Slide Time: 08:34)



So let me make some definitions regarding polynomials so there is something called total degree of a polynomial. So let $p(X_1,\ldots,X_n)$ be $\sum a_{(i_1,\ldots,i_n)} X_1^{i_1} \ldots X_n^{i_n}$ and it is a finite sum all but finite limited terms, all but finitely meaning this coefficients is are 0. So let this be polynomial in n variables over a field so if field like F could be like you know it will be set of reals rational or some finite fields F of Q, for Q equal to some prime to the point.

So it is where this coefficients are coming from the constants are coming from. Now we will define the total degree of a monomial each such term is called a monomial. So define total degree of $X_1^{i_1} \ldots X_n^{i_n}$ to be $i_1 + i_2 + \ldots + i_n$. The total degree of a polynomial is the highest total degree of its monomials.

(Refer Slide Time: 12:08)

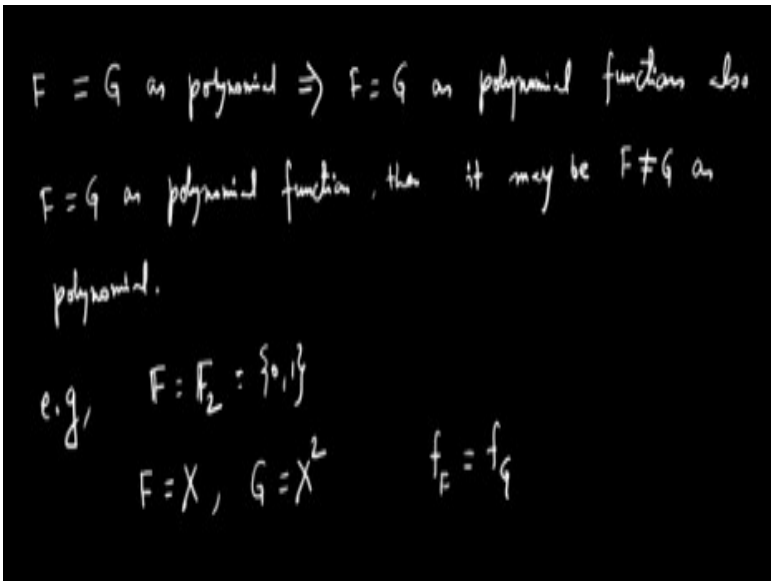


So for example let us take a polynomial maybe $10X_1^7X_2^3-5X_2^6X_3^5+19X_1^2X_2^6X_3$. So the total degree of the first monomial is 10 total degree of second monomial is 11 total degree of third monomial is 9 so the total degree of p this polynomial is 11. Now each polynomial we can associate with a polynomial function.

So what is a polynomial function? So let $p(X_1,\ldots,X_n)=\sum_{(i_1,\ldots,i_n)\in\mathbb{N}^n} a_{(i_1,\ldots,i_n)}X_1^{i_1}\ldots X_n^{i_n}$ with this let this be a polynomial. Then see a polynomial in n variables $X_1,\ldots,X_n$ this naturally defines in variables over field F. Then p defines a polynomial function let us call it f of p from F^n to F.

How does it define? f of $p(x_1,\ldots,x_n)$ where $x_1,\ldots,x_n$ are belongs to the field this is simply you put in the variable X_1 you put x_1 up to X_n put small x_n, $p(x_1,\ldots,x_n)=\sum_{(i_1,\ldots,i_n)\in\mathbb{N}^n} a_{(i_1,\ldots,i_n)}x_1^{i_1}\ldots x_n^{i_n}$.

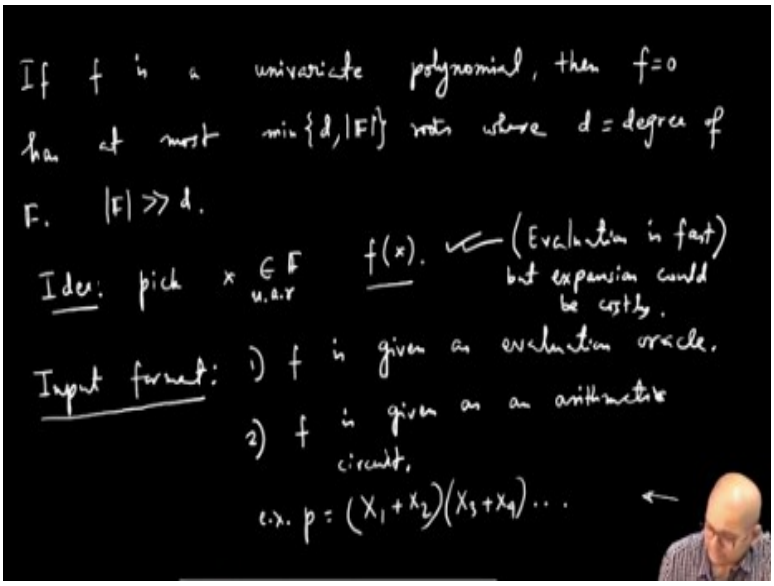
(Refer Slide Time: 16:01)



So now what is the algorithm so this is now we can have so if F = G, as polynomial then F = G, as polynomial functions also. But if F = G, as polynomial functions then if it is possible to have it may be F not equal to G, as polynomial. For example if you look at if you take the field to be F 2 where there are 2 elements basically 0 and 1 and addition and multiplication is done modulo 2.

Then if you take the first polynomial as X and the second polynomial as X square you know as polynomial function F; F = F G because both mapped 0 to 0 and 1 to 1 but as polynomial they are different. In polynomial identity testing problem we want to know whether they are same as polynomials. So whether F equal to whether F a given polynomial is identical to the 0 polynomial. So for that we need to prove important lemma which is called Schwentz - Zipple lemma so the let me first explain the high level idea.

(Refer Slide Time: 18:35)



So if f is univariate polynomial it means there is only one variable in f on only x, say then f = 0 has at most minimum of d, d is the degree and size of the field most d roots, where d is the degree of. So if the field is sufficiently large field is if size of F is much more than d then there del F is d + 1. So if the size of the field has much more than d then if I pick a uniformly random element from f it is most likely that it is not a 0 because there are only d many zeros.

So the randomized algorithm could look like that you know so the random the idea pick x from F uniformly at random and you know and evaluate this just f of x. Now if f of x is 0 output 0 and if f of x is not 0 then then say if f is not 0. So it seems that we have so it the input format becomes important how is the input is given input format? So 2 ways input could be given one is that f is given as evaluation oracle, f could be given as an evaluation oracle. You supply some element from the field and you get the f of x, you supply x and get f of x.

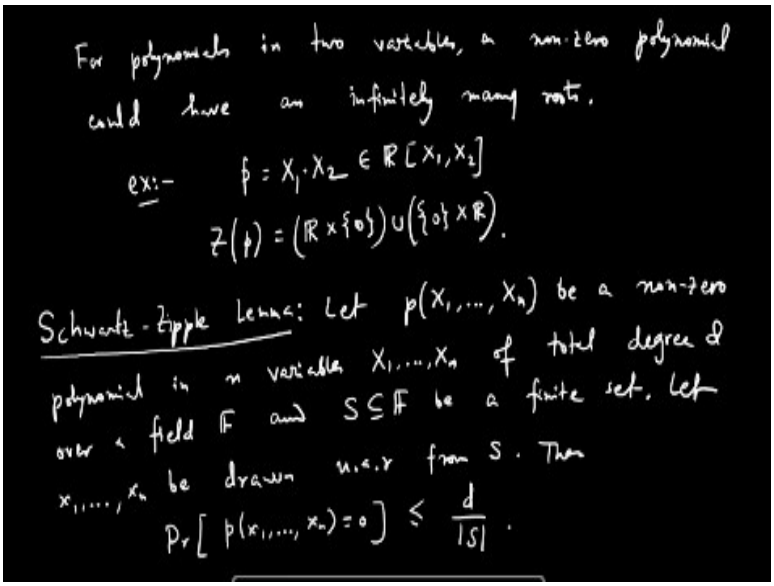
This other could be F is given as a circuit, F is given as an arithmetic circuit now and so one natural algorithm to check whether f is 0 or not is to write it in its canonical form that means write F in this form. That $\sum_{(i_1,\ldots,i_n)\in\mathbb{N}^n} a_{(i_1,\ldots,i_n)} x_1^{i_1}\ldots x_n^{i_n}$ if and this polynomial will be 0 if each of this term cancels out. But evaluating this writing on any arbitrary polynomial like this form could be expensive it can take exponential time.

For example you take this polynomial p equal to say X_1+X_2 if it is given in this product form say X_3+X_4 and so on. Now if you expand this it can it will have exponential many number of terms

in terms of the number of input size, so that straight forward method takes can take exponential time. So this sort of idea works for univariate polynomial that, take a uniformly random element from the underlying field and evaluate it.

So even if F is given in the circuit form or any succeed form you can put you can evaluate it at a, at a point and you can check whether this value is 0 or not. So that is evaluation is easy should be right evaluation is fast but expansion may be costly. Now you see that if you try to push this idea for multivariate polynomials we see a problem.

(Refer Slide Time: 24:51)



In for multivariate polynomials for even 2 variate polynomial for polynomials in 2 variables non-zero polynomial could have an infinitely many roots. So for example you know if say take $p = X_1 X_2$, so the solution set 0 of p is and suppose this is a real polynomial $\mathbb{R}[X_1, X_2]$ this is r cross 0 union 0 Cartesian product $\mathbb{R}$ which is an infinite set. But the idea of Schwentz - Zipple lemma; lemma is that you know the same technique somehow works.

So let me state Schwentz-Zipple lemma which we will prove in the next class. Schwentz- Zipple lemma that still even in this polynomial if we pick a field element for x_1 uniformly at random and a field element for x_2 uniformly at random and evaluate it with high probability it will be a non-zero value. So let me state let $p(X_1, \ldots, X_n)$ be non- zero polynomial in n variables $X_1, \ldots, X_n$ of total degree d over F over field f over a field if and is subset of if be a finite set.

Let $x_1,\ldots,x_n$ be drawn uniformly at random from s then probability that $p(x_1,\ldots,x_n)=0$ is at most d by cardinalities. So in the next lecture we will prove this and once we have this tool we will use this to design a Monte Carlo type randomized algorithm for polynomial identity testing problem so we will stop here.