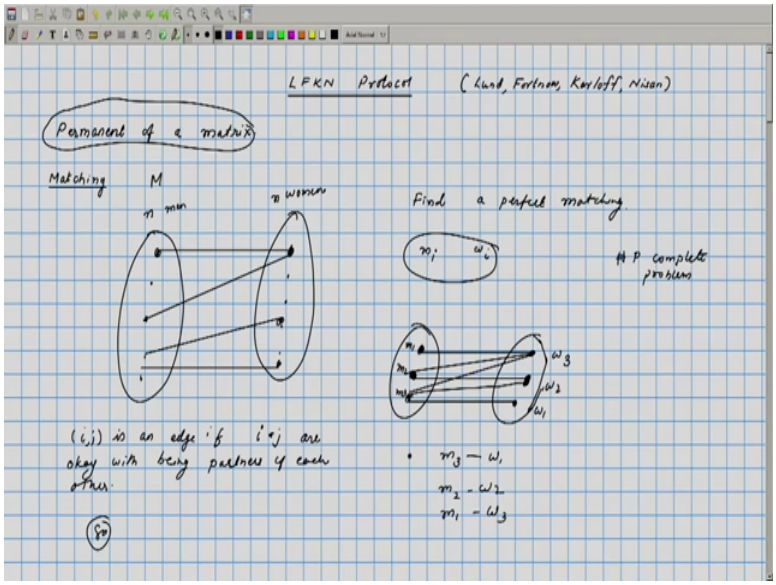
Randomized Algorithms
Prof. Benny George Kenkireth
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati

Lecture – 39
LFKN Protocol

(Refer Slide Time: 00:38)



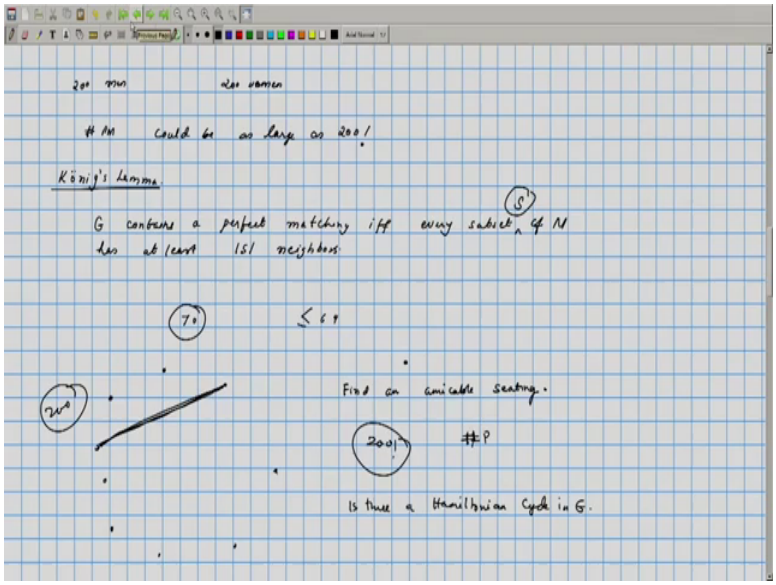
So, in this lecture we will learn about the LFKN Protocol. So, protocol is named after the inventor is the protocol Lund Fortnon Karloff and Nisan, the question is about calculating the permanent of a matrix. So, we will say how using this protocol one can compute the permanent of a matrix, before we get to that we will look at this problem more carefully; and see how it is related to some core computational issues.

So, let us just start with the small problem. So, let us look at the problem of matching ok. So, here there are n people ok, n men let us say and n women and we have this edge between a pair of people, if this particular person has interest that in the other person. So, i comma j is an edge if i and j are okay with being partners of each other. So, a natural question would be can we find a perfect matching ok; that means, find an arrangement. So, let us say m 1 or m i w i; so pair people in such a way such that m i w i is an edge ok. For example, if we take the, if you take this particular diagram whether 3 people; 3 men and 3 women and the following edges are present.

Now, the vertices if we name them as m 1, m 2, m 3 w 1 w, 2 and w 3 they only perfect matching in this particular graph is m 3, w 1 m 2 w 2, and m 1 w 3 this is the only perfect matching in this particular graph. We could also ask how many perfect matching's are there ok. So, in this there is only one perfect matching where as given arbitrary by part 8 graph finding the number of a perfect matching might be difficult that is known to be a sharp P complete problem ok, which means its a very difficult problem as a sharp P as a complex state class.

And computing the number of perfect matching's is a complete problem for that particular class. Let us look at the following question; suppose there are no perfect matching's in this how do we convince somebody that there is no perfect matching's?

(Refer Slide Time: 04:52)



So, the total number of perfect matching's let us say if there were 200 men and 200 women, the number of perfect matching's could be as large as 200 factorial ok. So, if you wanted to say that there are no perfect matching, we could list down all the possible matching's and argue that none of them are perfect matching's, but that is prohibitively expensive, because 200 factorial is not a small number.

Now, if there were no perfect matching's is there a way to convince some one that look there are no perfect matching's one could do that via Konig's lemma. So, Konig's lemma gives an if and only if condition for existence of perfect matching ok; G contains a perfect matching if and only if every subset S of N has at least size S neighbours ok.

So, it means if there are no perfect matching's then there is a subset, whose number of neighbours is less than the size of the subset ok. So, in order to show that there are no perfect matching's one has to just identify a subset; such that, the number of neighbour is strictly less than the size of the subset. So, if you find the 70 sized set such that its neighbours is less than or equal to 69, then we know that look these set of vertices cannot be matched at all they cannot all be simultaneously matched.

So, this does not mean that finding that set is easy, but it says that if there were no perfect matching then of course, there is one set there is at least one set; such that its neighbours are strictly less than this size of that is it. So, one can convince another person that there are no perfect matching's ok. So, if you assume that you are infinitely powerful in terms of computation, then you can convince anybody that a given graph does not have a perfect matching. If it really did not have a perfect matching we will do that by just figuring out which is the subset which does the job.

Now, let us look at a related problems so here we wanted to match amongst two set of vertices, you had to match m to w, suppose we had a problem of matching within the same set. In other words suppose you have n vertices and let us say here two people are connected by an edge is they do not mind being friends of each other ok. So, if you have an edge; that means, they are compatible; oh now we want to find a way in which these people can be seated round a table such that, everybody is next to there I mean nobody is sitting with somebody who is not there friend.

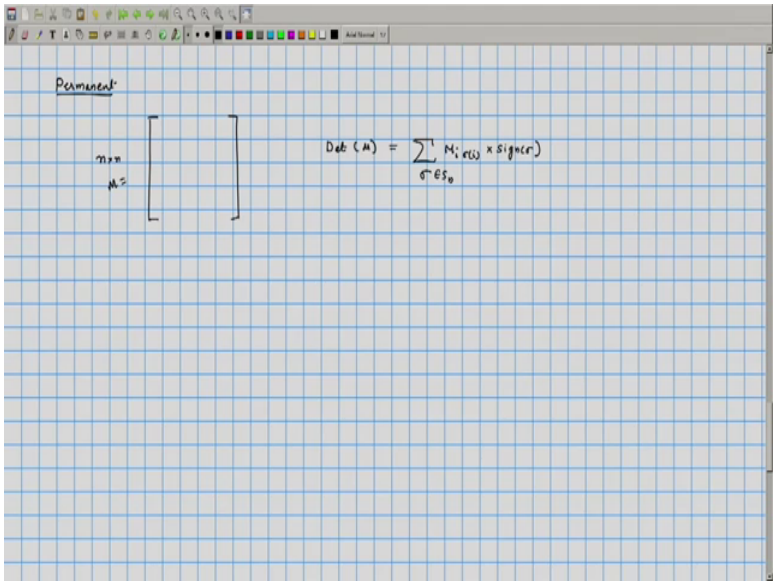
So, find an amicable seating, an amicable seating would essentially mean that your neighbours on the round table are your friends ok; can this be done again if there were 200 people here there are 200 factorial possibilities, you could try out all possibilities, but suppose there are no such possibilities that is there are no Hamiltonian cycles. So, this is a question of is there a Hamiltonian cycle in G, if there is one can of course, convince somebody by actually displaying the cycle.

If there are none then we need to look at all these 200 factorial possibilities, is there some way some clever argument like we had via Konig's lemma; in case of perfect matching that there cannot be any Hamiltonian cycle ok. So, this is an example of a problem that is in sharp P. So, how does one solve this problem? So, here in case of perfect matching it was easy to see that we could convince that there are none, but if

there are let us say 80 perfect matching's how do we convince somebody that there are 80 perfect matching's ok.

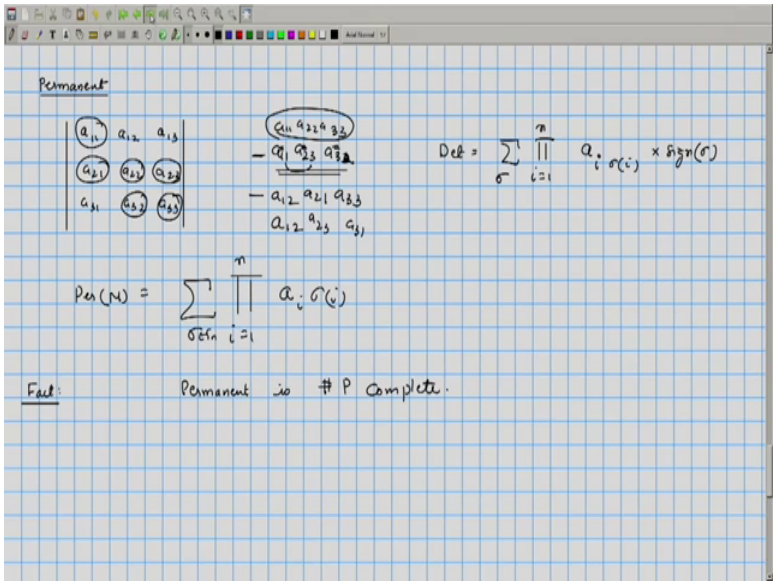
So, the number of perfect matching's is related to some quantity known as permanent of a matrix, It can be shown that the number of Hamiltonian cycle is also related to the permanent of a suitably constructed matrix. If the number of Hamiltonian cycles is 0 the permanent of an associated matrix will be 0. We have not yet defined what the permanent of a matrix is and that is what we will do in the next slide ok.

(Refer Slide Time: 10:36)



So, we will first defined what is a permanent of a matrix? The permanents are defined for square matrices. So, consider an n cross n matrix its determinant is written in the following way. So, determinant of M is summation overall permutations M i sigma i; now this is another cryptic notation we will just do this for a three cross three matrix this into sign of sigma. So, determinant of a matrix is summation over all permutations sigma so sigma belonging to S n m i sigma i times sin sigma.

(Refer Slide Time: 11:25)



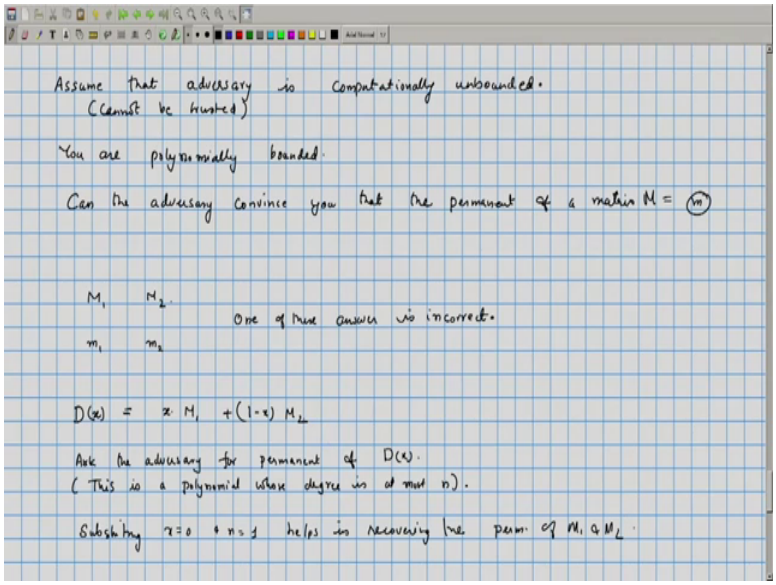
So, let us first look at a 3 cross 3 matrix, a 21, a 22, a 23, a 31, a 32, a 33; the determinant of this matrix would be a 11 into a 22 into a 33 that is the first term. The second term would be again a 11 and then a 23 and a 32 and this has a negative sign. The next term would start with minus a 12 then a term from the second row that is a 21 and the term from the third row a 33.

The next term again would be with the plus sign a 12 and the term from the second row a 23 a 31 ok. So, the determinant can be written as summation over products so sum is over all permutation sigma and product going from i equals 1 to n, for n cross n matrix; and each term has n terms at each term was of the from a i and sigma i. So, here a so i is going from 1 to 3, a 1 a 2 and a 3 and the so, this is a 23 into a 32, this into the sign there is a sign associated with each permutation and that is the formula for the determinant.

If you ignore the sign whatever you get is called as a permanent. So, permanent of a matrix M is equal to summation overall permutations product i going from 1 to n, a i sigma i; where sigma is any permutation on n length. So, we want to compute this particular quantity although determinant computation is very easy, you can do Gaussian elimination and compute the determinant. There are no easy algorithms known for computing the permanent its known that computing permanent is a sharp P complete problem.

And just say that permanent is sharp P complete ok, its also means that if you can compute permanent you can count the number of perfect matching's in a graph you can count the number of Hamiltonian cycles in a graph and so on ok. So, here in this lecture we will see protocol to compute the permanent of a matrix; there are some difference in this computation and a standard algorithm, in the sense here we will think of this as an interaction between two people ok. So, when we say a protocol what we have in mind is the following, we will make the following assumptions.

(Refer Slide Time: 14:39)



Assume that, the adversary there is another think of this is a two player game you are one player and there is the second player, and the other we will assume that the adversary or the other player is computationally unbounded. So, we can do any computations very quickly, but cannot be trusted in the sense the adversary might try to cheat you. He is capable of doing computations quickly, but he might not convey the answers to you faithfully; on the other hand you are polynomially bounded. In other words you can do only a polynomial amount of computations in a at a given time ok. Now, the objective is we want to answer this question can the adversary convince you that the permanent of a matrix M is equal to small m ok?

So, that is a given matrix, that is common for you and the adversary; and the adversary some how quickly figures out what is the permanent. And now the adversary wants to convince you that it is indeed m; you could try and do compute at yourself, but that is

going to be an exponential time computation it is not clear how permanents can be computed quickly. So, now, can the adversary convince you that the permanent is indeed small m ok; so we will see that randomization helps in this particular task ok. So, we will begin with the following simpler problem.

So, let us say that there are two matrices M 1 and M 2 ok. And the adversary has said that the permanent of M 1 is small m 1 and the permanent of M 2 the matrix m 2 is small m 2, one of these answers is incorrect or you suspect that it is incorrect ok. You are not sure, but you think that at least one of them is incorrect, now is there a way by which you can force the adversary to commit to the permanent of a single matrix whose value has to be incorrect with high probability ok. So, here your in a situation where you think that one of these matrices is incorrect, one of these answers is incorrect one of these permanents is an incorrect value.

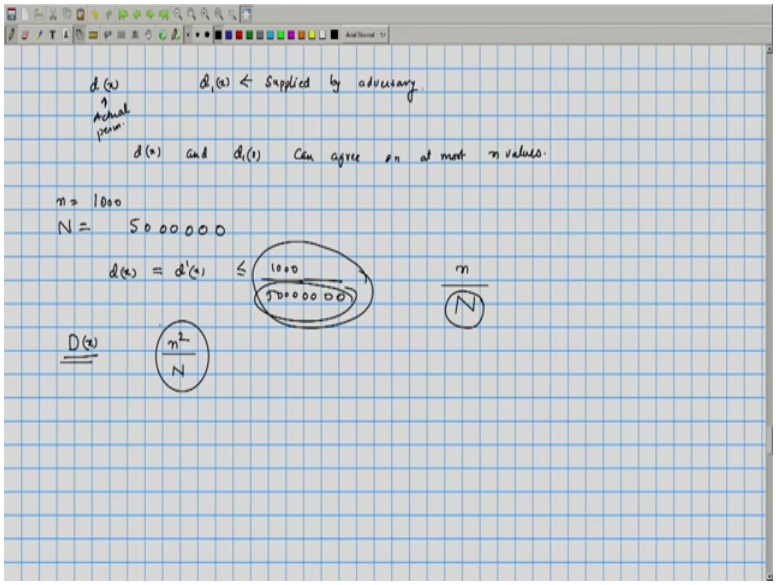
Can you somehow force the adversary to give a single matrix? Whose permanent is falsely stated we will see that can be done ok. So, what you will do is once you got this m 1 and m 2, you will think that the following matrix which let me call it as D x that x is a variable. So, this is a matrix x times M 1 plus 1 minus x times M 2; we will assume that M 1 and M 2 are both n cross n matrices ok. So, this means take the matrix M 1 multiply every term with x you will get a matrix with a variable at each position and you take the matrix M 2 multiply it with 1 minus x and this new matrix that you get you call it as D x.

Once you fix the value of x you will get a single matrix, but for different values you have different matrices. Now, D is the matrix whose entries are linear polynomials, so you can ask the adversary for the permanent of D x. So, that is the first thing that you would do ask the adversary for permanent of D 1 of D x. Now the permanent of D x is not a number, but it is a polynomial of degree at most n. So, the permanent is a polynomial whose degree is at most n.

Now, think about the following problem, if the adversary had stated the values of the permanent of M 1 and M 2 incorrectly, even if he has stated one of them incorrectly can he afford to give the correct value of the polynomial D x? If he gives the correct value of the polynomial D x you can substitute x equals 1 and recover the correct value of the permanent of M 1; and if you substitute x is equal to 0 you will get the correct value for the permanent of M 2 ok. So, substituting x equals 0 and 1 helps in recovering the

permanent of M 1 and M 2; if the adversary has lied in the values of the permanent of M 1 and M 2 he has to lie in the permanent of d, because if he correctly states the permanent of d then from that you can find that the adversary had lied. So, now, if you states an incorrect polynomial can we somehow figure out that he has lied that is also easy.

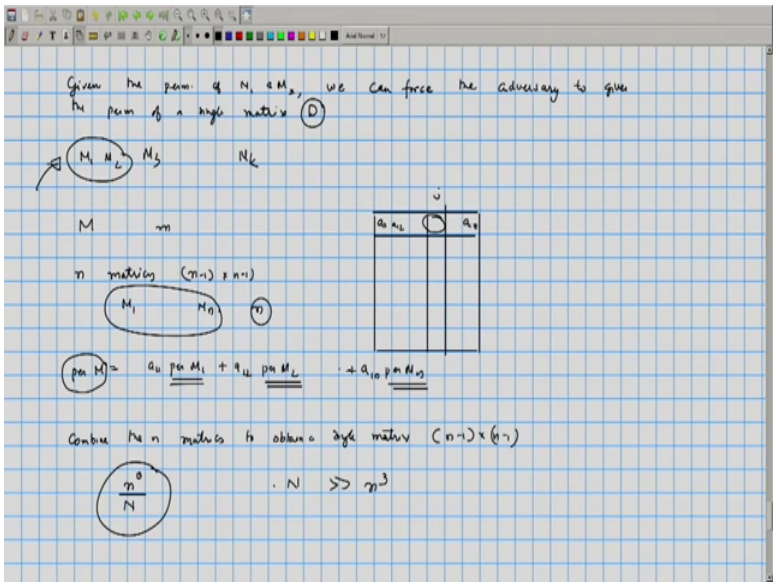
(Refer Slide Time: 20:51)



Suppose d was the actual polynomial and the adversary if he has not given the correct polynomial let us call it as d 1 x. So, this is supplied by the adversary note that d x is the actual one ok. So, d x and d 1 x can agree on at most n points. So, suppose small n was something like let us say 1000, you choose capital N is equal to let us say 50000 or 5000000, the probability that d x is equal to d 1 x is going to be less than or equal to 1000 by 50000 or 5000000 ok.

So, if he has incorrectly stated the value of one of the permanents we are forcing him to incorrectly state the permanent of the matrix D x for some random value x and he can be successful in hiding his lie only with a very small negligible probability ok. So, we had the situation that the adversary had lied about one of the permanents, we force him to lying on the value of a single matrix capital D ok. So now, how can we use this fact to have a protocol for computing the permanent of an arbitrary matrix?

(Refer Slide Time: 22:53)



So, let us just write down what we have achieved given the permanent of M 1 M 2 we can force the adversary to give the permanent of a single matrix say D if the permanent of M 1 and M 2 are incorrectly stated with very high probability the adversary will have to lie about the permanent of D ok. So, if he had a collection of matrices M 1 M 2 M k we can do this k times, we can first combine M 1 and M 2 in this particular manner and that result you can combine with m 3 and so on.

So, even if we are given k matrices and if the adversary had lied in even one of those permanents, we can force him to give the permanent of a single matrix, if you are given M matrices and the adversary has given the permanent of each of those matrices. If he has lied in even one of them we will catch the lie with high probability by forcing him to lie about the permanent of a single matrix ok. So, how do we use this to compute the permanent of a single matrix? So, let us say M is a matrix and the adversary has stated that its permanent is small m ok. Now we ask the adversary for the permanent of n matrices which are of dimension n minus 1 cross n minus 1 namely we look at the minor matrices.

So, if you have this matrix and if you expand along the first row corresponding to this particular entry; we will look at the matrix obtained by throwing out the first row and the j th column ok. So, there are n minus 1 such there are n such matrices we ask for the permanent of all those matrices. So, these m matrices if you call it as M 1 to M n and if

these values a 11, a 12 and a 1 n; we can ask the adversary to give the permanent of all these matrices M 1 to M n. If he has correctly stated the values of M 1 to M, n he cannot really be lying about the value of m because permanent of m is equal to a 11 into permanent of M 1 plus a 12 into permanent of M 2.

So, on up to a 1 n into permanent of M n, if the permanent of m was incorrectly stated then the adversary must give incorrect values for at least one of these matrices. If he has given the correct value for the permanent these matrices, we can do this consistency check of multiplying each of the permanents with the appropriate a i j's. And adding them up and we can compute the true value of the permanent.

So, if the true value of the permanent was incorrectly stated it forces the adversary to lie about the correct values of the permanent of the sub matrices. And so now, we are at this particular stage where instead of the permanent of a single n cross n matrix, adversary has given us the permanent n n minus 1 cross n minus 1 matrix out of which at least one of them is going to be incorrect. If he had correctly stated the permanent of M there is no problem.

If you has incorrectly stated the permanent of m than at least one of these smaller matrices then the adversary must given incorrect value for at least one of the smaller matrices. We can of course, combine them and get a single n minus one matrix whose permanent is incorrectly stated ok. So, we can combine the n matrices to obtain a single matrix, but now the dimension is n minus cross n minus 1. So, if the adversary was lied on a single matrix of size n cross n; we will force him to lie on a single matrix n minus 1 cross n minus 1. And this can recourse and in the end he has to lie about the permanent of a 1 cross 1 matrix.

But the permanent of 1 cross 1 matrix we can anyway directly find out and therefore, the adversaries lie can always be caught with very high probability. We argued that if you choose our n to be large enough higher than the degree of the polynomial, then we will get an error of small n the degree by capital N where capital N is the size of the universe from which we chose our random number. So, small n by capital N is the probability of error.

If we are doing this once, but we had combined n matrices into a single matrix and that would incur an error of n square by N. and then after combining n of these matrices we

are getting n recourse for another n step; so, overall if you work out the error probability that will be bounded by n cube by N.

So, your error probability is going to be strictly less than n cube by N and if you choose capital N to be greater than let us say n cube, then you know that the error probability is very small. And so, if of course, if the adversary had given the correct value each time; then we will verify all the steps we will see that all the consistency criterion are really met with and therefore, that causes no problem.

If the adversary lies even once about the permanent of a sub matrix, we can force him to propagate his lie. And, in the end he will get caught the probability that he does not get caught is bounded by n cube over n. So, that brings us to the end of the LFKN protocol.