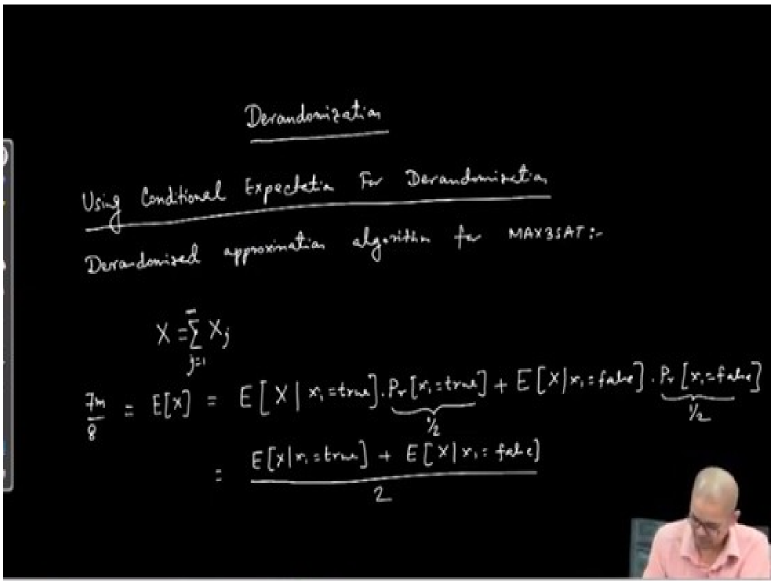
Selected Topics in Algorithm
Prof. Palash Dey
Department of Computer Science and Engineering
Indian Institute of Technology- Kharagpur

Lecture - 41
Derandomization

In the last class we have seen two randomized approximation algorithms. In this class, we will derandomize those algorithms.

(Refer Slide Time: 00:38)



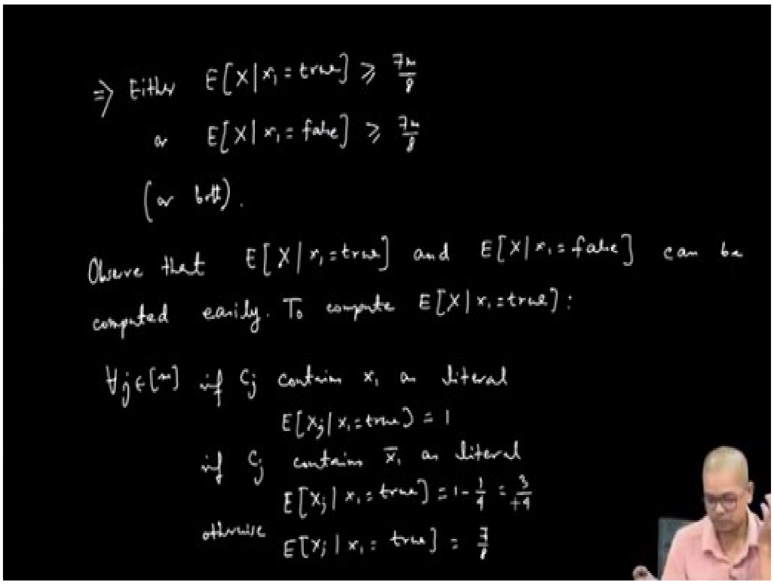
Derandomization. So derandomization is an umbrella technique to randomize an approximation algorithm. So it is not just one technique. So we will see one kind of derandomization using conditional expectation. So derandomization using conditional expectation for derandomization. For that we take the example of MAX3SAT. So derandomized approximation algorithm for MAX3SAT.

So let us recall we denoted X the random variable which is the number of clauses satisfied by your randomized algorithm. And for each clause, we had an indicator random variable X_j indicating whether to satisfied or not. And we had X is $\sum_{j=1}^{m} X_j$. And what is expectation of X? We computed it, it was $\frac{7m}{8}$. Now this is expectation of X can be written as you know there are n variables.

So we are setting each variable with true with probability half and false with probability half. This is expectation, let us take one variable say X_1 is true times probability that X_1 is true plus expectation of X when X_1 is false times probability that X_1 is false. So that means now each is half each of these probability is half. So this is expectation of X given X_1 equal to true plus expectation of X given X_1 equal to false by 2.

So hence $\frac{7m}{8}$ expectation of X is the average of these two conditional expectations. And this is $\frac{7m}{8}$.

(Refer Slide Time: 04:54)



That means either expectation of X given X_1 is true is greater than equal to $\frac{7m}{8}$ or expectation of X given X_1 equal to false is greater than or equal to $\frac{7m}{8}$ or of course both. Now here we observe that computing this expectation is easy. So there is a crucial observation. This technique can be is useful only if these conditional expectations can be computed easily.

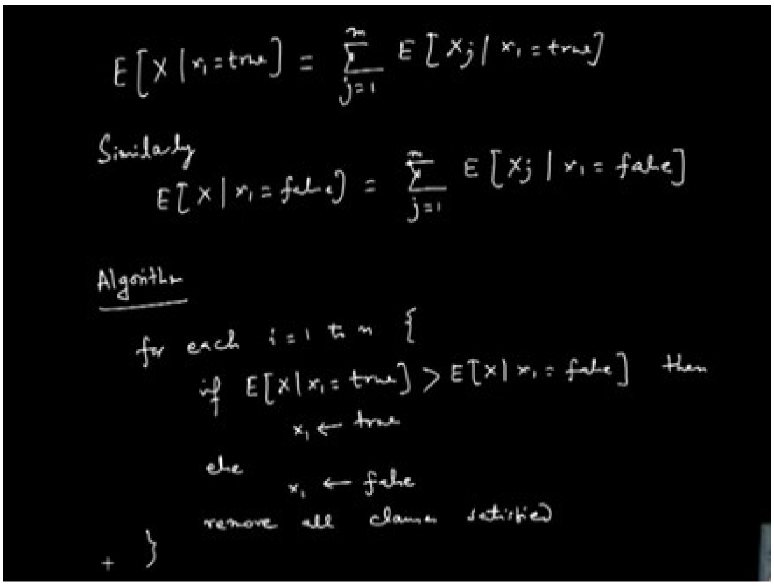
So observe that expectation of X given X_1 equal to true and expectation of X given X_1 equal to false can be computed easily, okay? So what is the idea, let us see. So if X_1 is true, so some set of clauses are satisfied. So let s see how it is computed for all X_j for all j in m if this clause C_j contains you know X_1 as literal. So we are discussing how to compute say expectation of X given X_1 equal to true.

Let me write, to compute expectation of X given X_1 equal to true, will do the following. So all those clauses which contain X_1 as a literal it is satisfied because X_1 is true. So expectation of X_j is 1. If C_j contains $\bar{X}_1$ as literal then it is not satisfied. Then this particular variable does not satisfy and the clause C_j could be satisfied using only the remaining two literals.

So expectation of X_j or let me write here X_j given X_1 is true is 1. Expectation of X_j in this case given X_1 is true is now the probability that it is not satisfied is both the remaining two literals are also turn out to be false that happens with probability $\frac{1}{4}$. So with probability $1-\frac{1}{4}$ this is satisfied. So this is $\frac{3}{4}$.

Otherwise you know if C_j does not contain X_1 or $\bar{X}_1$ then expectation of X_j given X_1 equal to true remains same. So now for each clause, we now know the expectation of X_j given the event that X_1 equal to true. And now we add all these things and hence that sum.

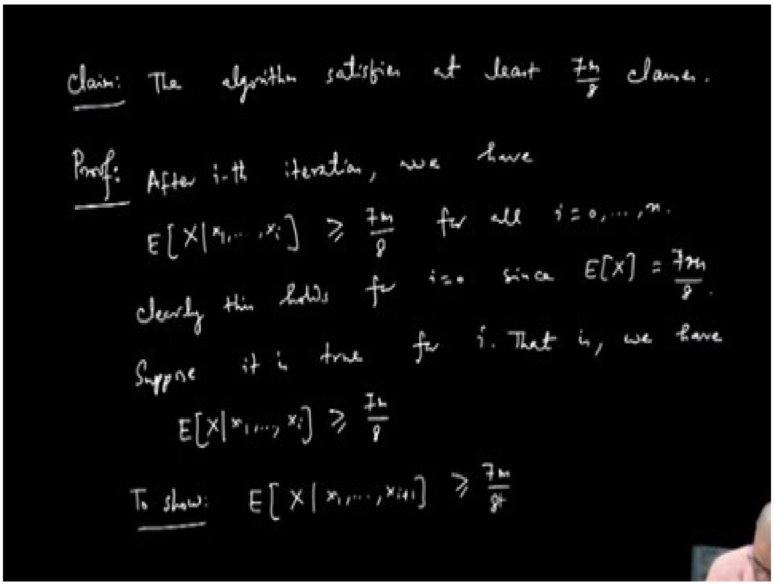
(Refer Slide Time: 10:01)



So because expectation of X given X_1 equal to true is what? This is sum of sum over j equal to 1 to m expectation of X_j given X_1 equal to true and we have easily computed what are these individual expectations and we add them and this way we can compute expectation of X given X_1 equal to true. Similarly expectation of X given X_1 equal to false can be computed in exactly the same manner, is also j equal to 1 to m expectation of X_j given X_1 equal to false.

So what is our derandomized algorithm? Now for each variable i equal to 1 to n if expectation of X given X_1 equal to true is greater than expectation of X given X_1 equal to false then set X_1 to be true else set X_1 to be false. Remove all clauses satisfied, okay?

(Refer Slide Time: 12:43)



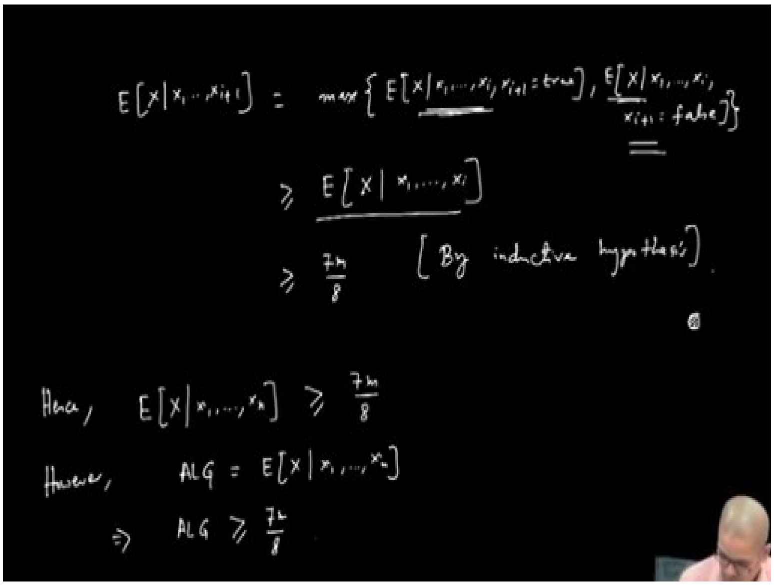
So you know, claim the algorithm satisfies at least $\frac{7m}{8}$ clauses, okay? Proof, so it is by loop invariant. So in i-th iteration we have expectation of X given an i-th iteration or if we write at the start of i-th iteration or after i-th iteration. After i-th iteration $x_1,..,x_i$ are set. So $x_1,..,x_i$ whichever it is set, this is greater than equal to $\frac{7m}{8}$. This is the claim. So this is for all i equal to 0 to n, okay?

So clearly this is, clearly this holds for i equal to 0 since expectation of X that we have computed we have already seen this is $\frac{7m}{8}$. So this is a proof by induction.

Suppose it is true for i, then I will show for i + 1. That is we have expectation of X given $x_1, \ldots, x_i$. This notation means that in whichever way $x_1, \ldots, x_i$ have been set by the algorithm. So this is greater than equal to $\frac{7m}{8}$.

To show this is the inductive step expectation of X given $x_1, \ldots, x_{i+1}$, this is greater than equal to $\frac{7m}{8}$.

(Refer Slide Time: 16:28)



So what is the expectation of X given $x_1, \ldots, x_i$, this is expectation of X given $x_1, \ldots, x_{i+1}$ is set to true times probability, sorry this is write x_{i+1}, this is max of expectation of X of $x_1, \ldots, x_i$. They have been set before and x_{i+1} is set to true. That is what the algorithm does. In the i + 1-th iteration it sets x_{i+1} to true and false and see whichever is more, whichever expectation is more.

That way it is set. So this expectation of X given $x_1, \ldots, x_i$ and then x_{i+1} is false. So this is you know because expectation of X given $x_1, \ldots, x_{i+1}$ is the average of these two expectations. So this is less than equal to or is equal to sorry, equal to or this is greater than equal to, this max is greater than equal to expectation of X given $x_1, \ldots, x_i$

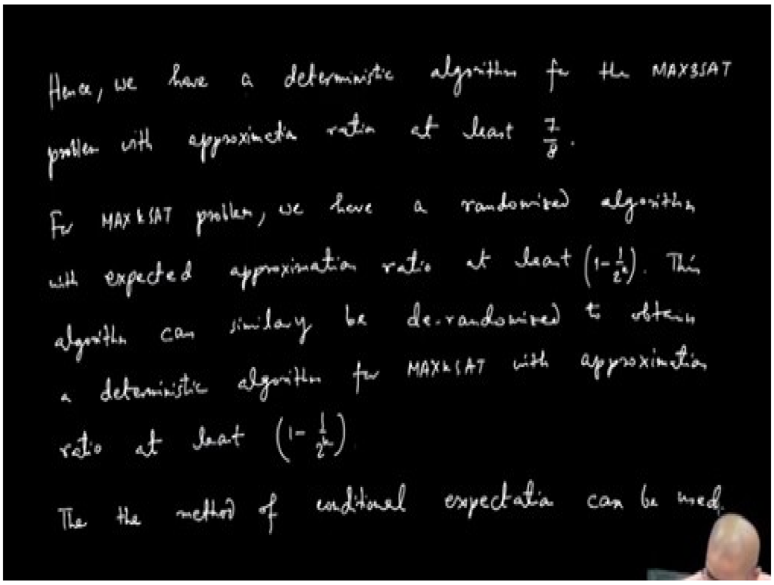
.

Because expectation of X given $x_1, \dots, x_i$ is the average of expectation of X given $x_1, \dots, x_{i+1}$ set to true and expectation of X given $x_1, \dots, x_{i+1}$ is set to false. It is expectation of X given $x_1, \dots, x_i$ is the average of these above two things. And so it is max, max is greater than equal to this. But this is, by inductive hypothesis this is greater than equal to $\frac{7m}{8}$. This is by inductive hypothesis.

This proves the claim. Hence we have expectation of X given $x_1, \dots, x_n$ is greater than equal to $\frac{7m}{8}$. But what is X so X capital X expectation of X given $x_1, \dots, x_n$ that means all the variables has been set and this is one number and this is nothing but ALG. This is but however ALG the number of clauses satisfied by the algorithm is nothing but expectation of X given $x_1, \dots, x_n$.

Hence, we have ALG is greater than equal to $\frac{7m}{8}$. Hence, we have a deterministic algorithm. You know in this algorithm, there is no randomness involved. We are not making any random coin tosses. We are comparing this to expectations using this framework and we make deterministic choices.

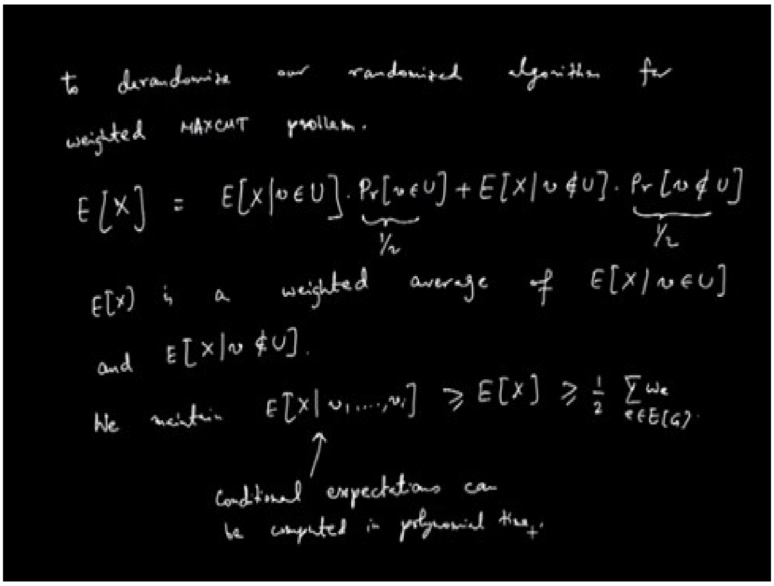
(Refer Slide Time: 21:16)



Hence we have a deterministic algorithm for the MAX3SAT problem with approximation ratio at least $\frac{7}{8}$. You know using the same approach we can see that for MAX3SAT set problem where each clause has key min literals for MAXkSAT problem we have a randomized algorithm with expected approximation ratio at least $\frac{7}{8}$.

Again this algorithm can similarly be derandomized to obtain a deterministic algorithm for, sorry for MAXkCut this approximation ratio is $\left(1-\frac{1}{2}\right)^k$. For MAXkCut MAXkSAT with approximation ratio at least $\left(1-\frac{1}{2}\right)^k$. So I encourage you to apply the same method of conditional expectation to derandomize our randomized algorithm for MAXCut.

(Refer Slide Time: 26:01)



So let me write here. The same method of conditional expectation can be used to derandomize or randomized algorithm for weighted MAXCut problem. And the idea is exactly same, you know here there also you look at x was the expected size of the cut and there also expectation of X can be written as expectation of X given a vertex v you know this vertex v this belongs to this random cut that we are constructing.

v belongs to u times probability that v belongs to u plus expectation of X given v does not belong to u times probability that v does not belong to u. And the same framework applies because these probabilities are half and even if actually these probabilities are not half because these probabilities sum to 1. The main idea is expectation of X is a weighted average. So this half is not necessary.

Any such probabilities would have worked. weighted average of expectation of X given v in u and expectation of X given v does not belong to u. And hence, in each step we iterate over vertex and put $v \in U$, if expectation of X given $v \in U$ is more than expectation of X given $v \notin U$.

So in each step we are putting we are getting a or making a decision which increases which always retain that expectation of X given the current whatever variables are set this at least half, this greater than equal to expectation of X. And expectation of X is greater than equal to sum of weights of the edges by 2.

And hence, so we maintain expectation of X given this setting $v_1, \ldots, v_i$, this is always greater than equal to expectation of X which is greater than equal to half of the sum of weights okay, $E[G]$. And here again one important thing to check is that these things can be computed in polynomial time. So this conditional expectations can be computed in polynomial time, okay.

And hence again the same approach can be used so I will let you fill up the details. So let us stop here today.