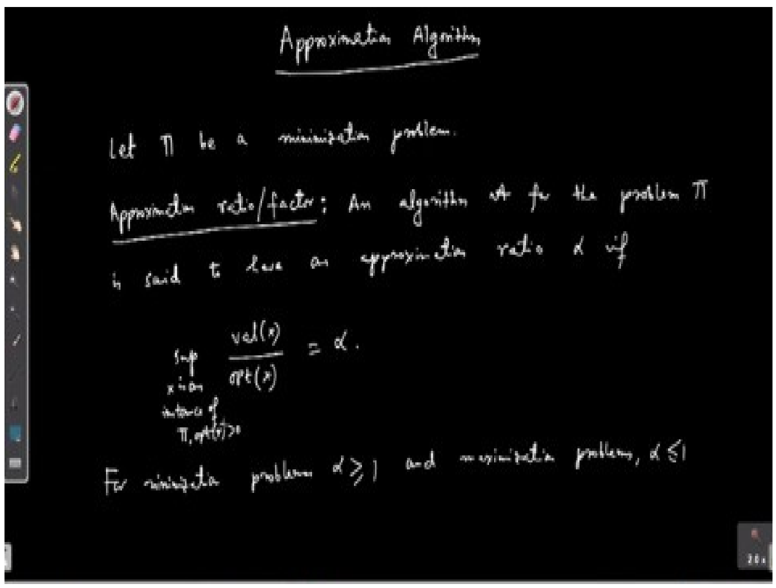
Selected Topics in Algorithm
Prof. Palash Dey
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 40
Randomized Approximation Algorithm

In the last class we have observed the equivalence of decision version and optimization version of a problem. So, in today's class we will start looking at approximation algorithms.

(Refer Slide Time: 00:38)

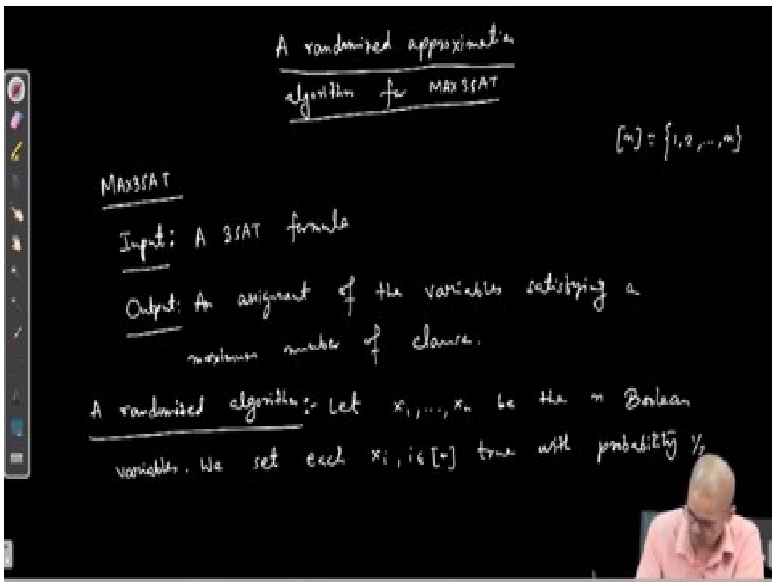


So, here we will be working with optimization version of problems. So, what is an approximation algorithm for that? Let us focus on say minimization problem, so let Π we are minimization problem that means what? That means we need to output a solution with minimum value and an algorithm so let us define a notion called approximation ratio or approximation factor. An algorithm A for the problem Π is say to have an approximation ratio α.

If for all instance val of x by no opt which is the optimal that means the minimum value of the solution of the instance is x and you maximize it take supremum over x is an instance of Π this is alpha. So, for maximization version with the exact opposite for maximization version we take supreme off opt over value. So, the motivation of approximation algorithm is for NP complete problems can we have polynomial time algorithms which has small approximation ratio.

Or let me write here for maximization problems so for minimization problems alpha is greater than equal to 1. Of course, because opt is the minimum value and let us for maximization works versions also let us define approximation ratio as value by opt so, for maximization problems alpha is less than equal to 1. This definition of course what if x is not equal to 0 opt of x is greater than 0.

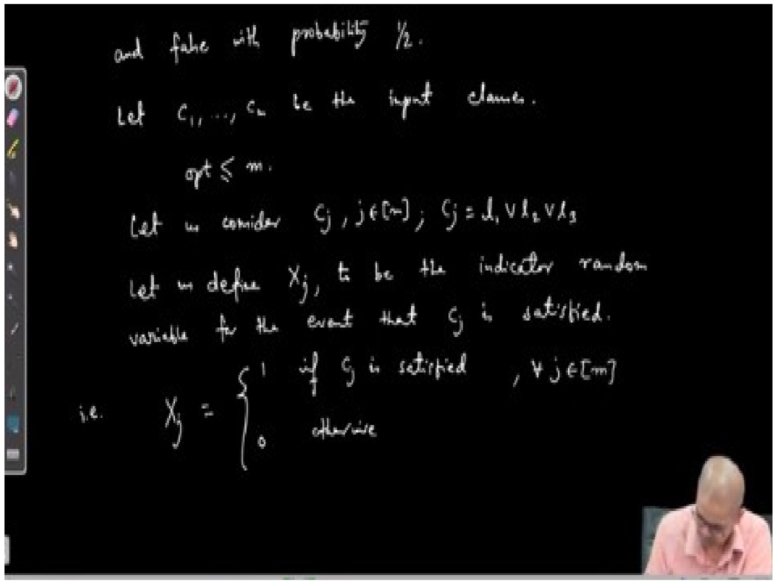
(Refer Slide Time: 05:08)



So, now let us see an example of an approximation algorithm. So, our first example is a randomized approximation algorithm for max 3SAT. So, max 3SAT is an optimization version of 3SAT. What is max 3SAT? Input a 3SAT formula output an assignment of the variables satisfying maximum number of clauses. So, we are looking for an assignment which maximizes the number of clauses we are trying to satisfy as many clauses as possible.

So, here for this we have a simple randomized algorithm so a randomized algorithm, how does it work? It simply so let $x_1, \ldots, x_n$ be the in Boolean variables we set each $x_i \in [n]$ this third bracket n this is the set for any natural number n this third bracket in is the set of all integers from 1 to n. We set each x_i true with probability half and false with probability half.

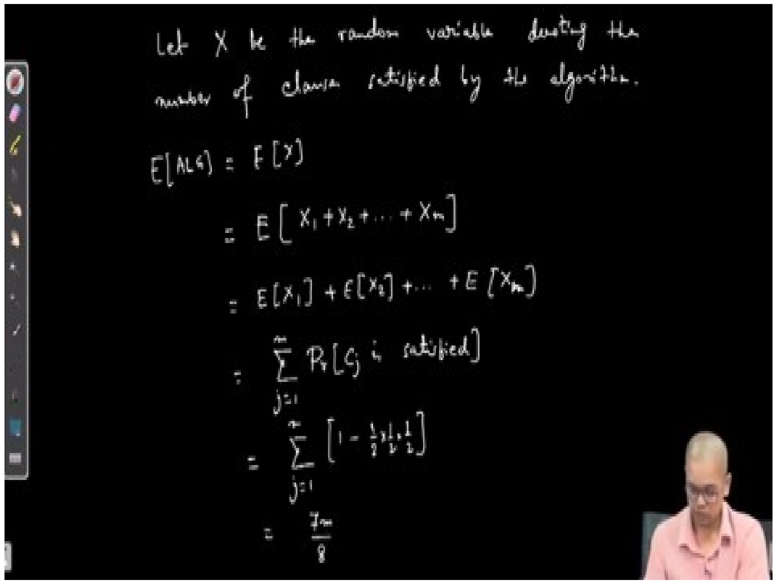
(Refer Slide Time: 08:53)



Let us see what is the expected number of clause satisfied. So, let $C_1, \dots, C_m$ be the input clauses so of course opt which is the maximum number of clauses that could be satisfied is less than equal to m, now let us see how many clauses that our algorithm satisfies because it is a randomized algorithm it is a Monte Carlo type randomized algorithm. We need to find we need to look at what is the expected number of clauses satisfied.

So, let us consider C_j for $j \in [m]$ let C_j be; because it is a 3SAT formula it is an or of three literals $l_1 \vee l_2 \vee l_3$. Let us define X_j to be the indicator random variable for the event that C_j is satisfied that means X_j, that is X_j is 1 if C_j is satisfied by the algorithm by the setting by a random setting of the n variables if C_j is satisfied and the 0 otherwise this is for all j in m.

(Refer Slide Time: 12:16)

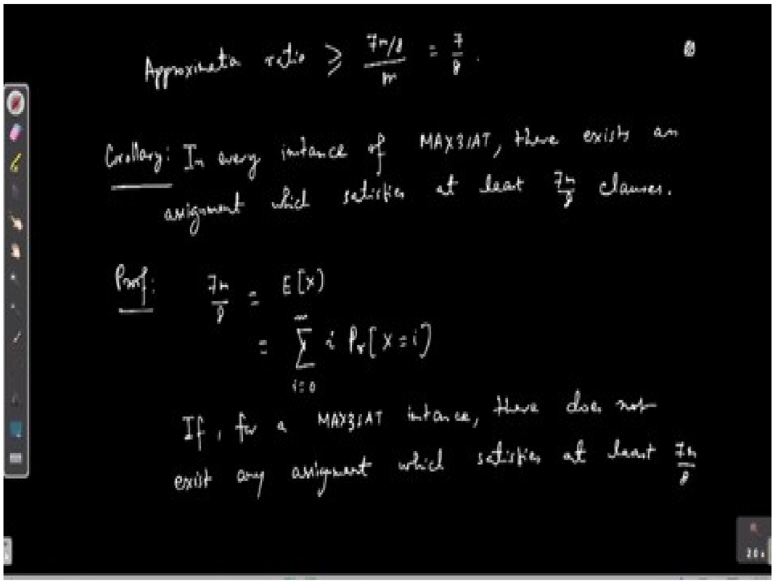


So, what is the and let X be the random variable denoting the number of clauses satisfied by the algorithm that means L so expected value of the solution of the algorithm that is L and expectation of L is expectation of x, now what is x in terms of $X_1, \ldots, X_m$? It is nothing but $\sum X_1 + \ldots + X_m$. From linearity of expectation, we get $\sum E[X_1] + \ldots + E[X_m]$.

Now what is expectation of each X_i? This is $\sum_{j=1}^{m} E[X_j]$ and because it is an indicator random variable expectation is nothing but probability of the event probability that C_j is satisfied. So, what is the probability that C_j is satisfied? This is the probability that it is 1 - the probability that C_j is not satisfied. The probability that C_j is not satisfied is the only way it cannot be satisfied is l_1 is set to false l_2 is set to false and l_3 is set to false.

Each happens with probability half so this is $1 - \frac{1}{2}\frac{1}{2}\frac{1}{2}$; this $\frac{7}{8}$ which is $\frac{7m}{8}$.

(Refer Slide Time: 15:18)

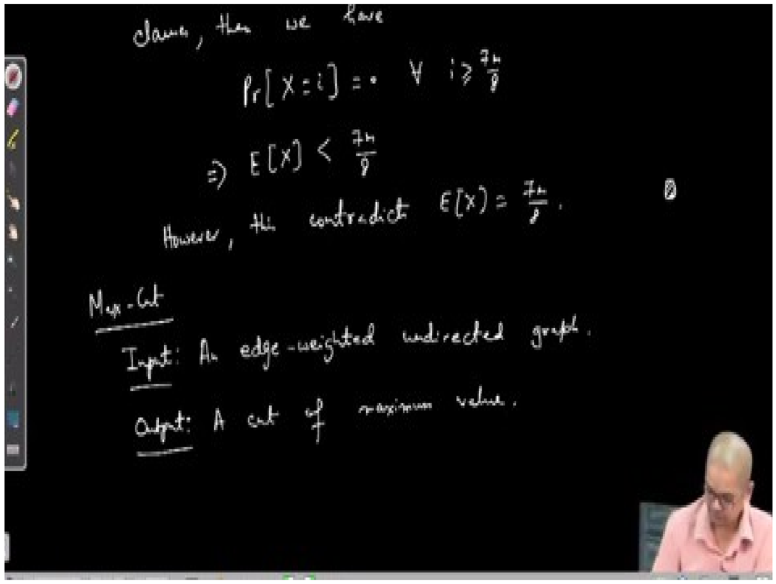


So, approximation ratio is greater than equal to now expected algae is $\frac{7m}{8}$ and up towards at most one that is why approximation is ratio is greater than equal to, at most m less than equal to m so approximation ratio is greater than equal to $\frac{7}{8}$. So, we have a $\frac{7}{8}$ randomized approximation algorithm for max 3SAT problem. Now couple of observations first of all we can derive a nice corollary from here, that the expected number of clauses that our algorithm satisfies is $\frac{7m}{8}$.

And what is this, in this is expectation of X on the other hand expectation of X can also be written as you know let me write first. For every instance of max 3SAT or we write max 3SAT there exists and assignment which satisfies at least $\frac{7m}{8}$ clauses. Why? You know, what is $\frac{7m}{8}$? $\frac{7m}{8}$ is expectation of X. Now what is expectation of X? This is X can take X is the number of clauses that is satisfied by the algorithm.

So, it can take value from say 0 to m this is i times probability X = i this is from the definition of expectation. Now for contradiction if for max 3SAT instance there does not exist and any assignment which satisfies at least $\frac{7m}{8}$ clauses.

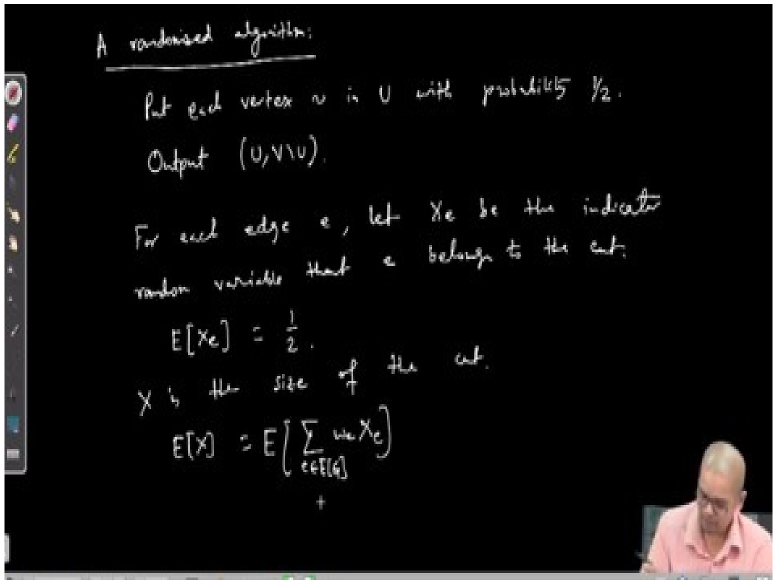
(Refer Slide Time: 19:34)



Then we have probability that X = i to be 0 for all i greater than equal to $\frac{7m}{8}$. Now that means in this sum all the probability values where for i = $\frac{7m}{8}$ to m, the m they are 0 that means this implies that expectation of X so, expectation of X is then convex combination of some number from 0 to less than $\frac{7m}{8}$. And hence this then expectation will be strictly less than $\frac{7m}{8}$. But this contradicts however this contradicts expectation of X is $\frac{7m}{8}$.

The same approach can be adopted same algorithm can be erupted for max cut problem. So, what is max cut problem? Input and edge weighted undirected graph. Output, a cut of maximum value, what is the value of the cut? It is the sum of the weights of the edges crossing the cut.

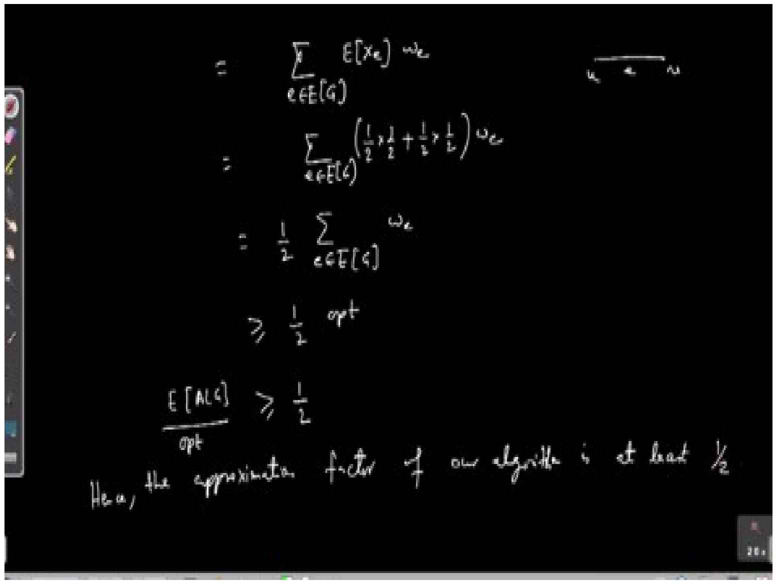
(Refer Slide Time: 22:22)



So, again what is the randomized algorithm it is a two half factor randomized algorithm put each vertex v in say U with probability half output U, V – U. What is the expected size of the cut? So, again for each edge e let X_e be the indicator random variable that e belongs to the cut. So, that means expectation of X_e is half, X is the size of the cut. So, expectation of x what is the size of the cut?

For all edges $e \in E[G]$ what is the contribution of this edge in the cut. If this cut if this edge belongs to the cut that is X_e then its contribution is its weight w of E so, X is summation $w_e X_e$. And we apply expectation.

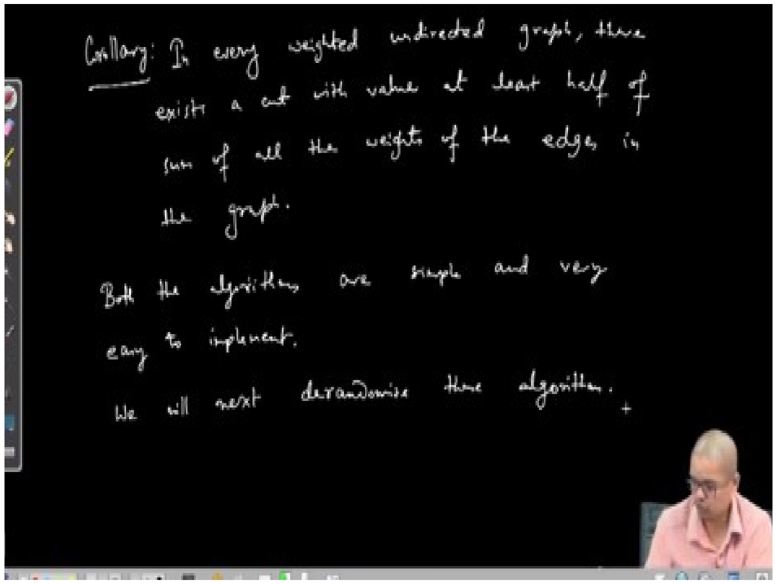
(Refer Slide Time: 25:04)



Now we apply linearity of expectation, using linearity of expectation we have w_e summation e in E, E, G expectation of X_e. This each expectation is half what is expectation of X_e this is the probability suppose this edge e is between u and v. So, this will contribute to the cut in two cases u belongs to capital U and v does not belong to capital U that happens with probability half times half or u does not belong to capital U and v does belongs to capital U half times half.

This is sorry w_e should be inside w_e this is half of summation $e \in E[G]$ w of e sum of all weights of the edges. But some of all weights of the edges is greater than equal to opt this is half opt so, what we have is expectation of AL by opt is greater than equal to half. Hence the approximation factor of our algorithm is at least half.

(Refer Slide Time: 27:35)



And again, we can derive the same corollary, in every weighted undirected graph there exists a cut with value at least half of sum of all the weights of the edges in the graph. So, here also you see how randomization can help us in designing very efficient algorithm. So, both the algorithms are simple and very easy to implement. And in next couple of lectures, we will see more involved approximation algorithms.

And also, we will see techniques how we can de-randomize this using the same idea we can achieve the same approximation bound that means for max 3SAT and approximation ratio of $\frac{7}{8}$ and for max cut an approximation ratio of half with the deterministic algorithm using the same idea. So, that we will see, so that is the de-randomization. So, we will next de-randomize these algorithms so, this we will see in the next class. So, let us stop.