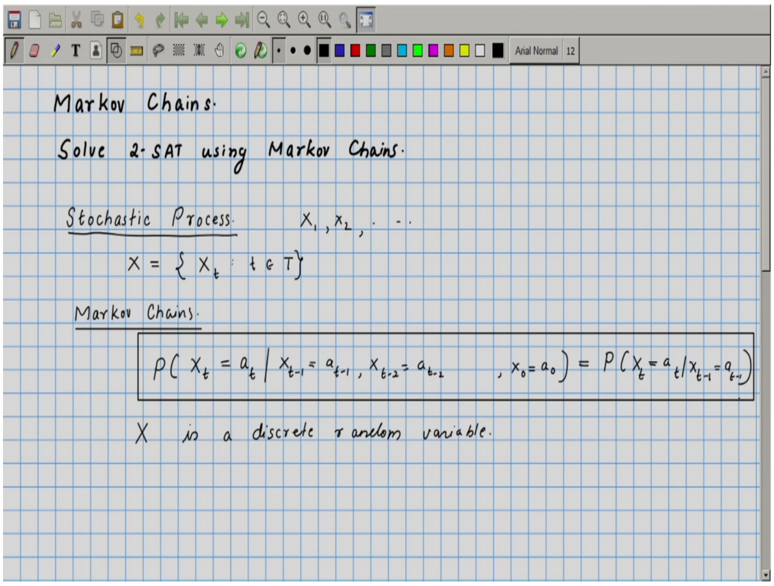
Randomized Algorithms
Benny George Kenkireth
Department of Computer Science & Engineering
Indian Institute of Technology, Guwahati

Lecture – 17
2-SAT and Markov Chains

So, in today’s lecture we will see what is called as Markov Chains.

(Refer Slide Time: 00:31)



We will use Markov Chains to design randomized algorithm to solve 2-SAT, ok. So, let us understand what are Markov Chains. So, first we will see what is a stochastic process. So, stochastic process is nothing but an indexed sequence of random variables. So, if you look at random variables X 1 X 2 so on this is what is in general as stochastic process.

So, we will write this as X, so X is a stochastic process we will write this as equal to X t; t belonging to some particular index set t. So, we will generally think of this small t as the time ok. So, we can think of, let us say observing the number of cars which pass a given point at given time t ok. So, we observe the vehicles passing and at integer times time t equals 0, 1, 2, and so on, and the number of cars passing at a given point at a time t that will be our random variable X t ok.

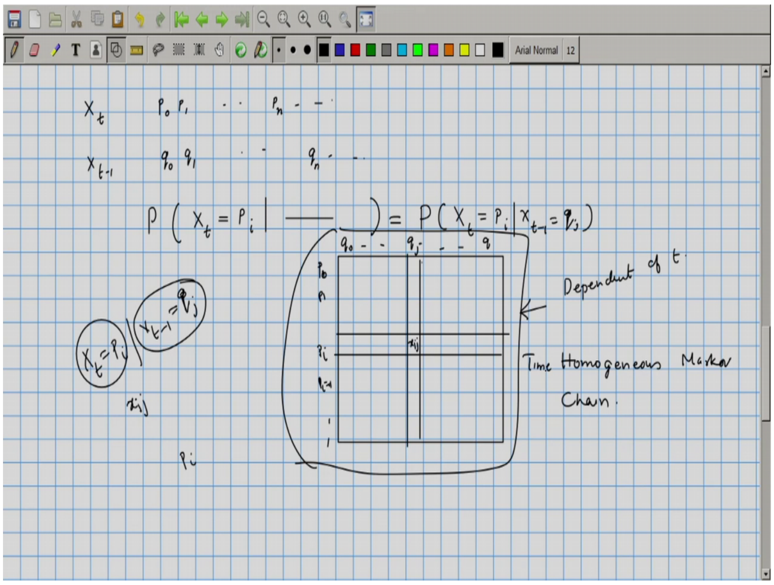
So, these are stochastic process. We are interested in particular type of stochastic process. So, these random variables X 1 X 2 etcetera there could be dependence amongst them, ok. Suppose we know that certain thing has happened in the value of the random variable X 5 is something, that might tell us about I mean tell us something about the random variable X 6 or X 7 so on. But we are interested in just some very specific kind of random process is which will be called as Markov Chains ok.

So, what is a defining property of Markov Chains? So, the probability that X t is equal to let us say a t, given all the information up till time t. So, we know what are the values of the random variables X t minus 1 so, let say this was equal to a t minus 1 and X t minus 2 equals a t minus a t minus 2. And X 0, let us say we start at time t equal 0 a 0. So, this probability is equal to probability that X t is equal to a t given X t minus 1 equals a t minus 1. So, this entire probability is equal to the probability that X t equals a t given X t minus 1 equals a t minus 1.

So, in some sense the entire dependence on the other random variables is captured by the random variable X t minus 1. So, the probability that the random variable at time t takes a particular value, given the entire history the values that the other random variables that is X 0 to X t minus 1; if their values or the history is known. Based on that if you try to compute the probability that X t equals a t that is exactly same as probability that X t equals a t given X t minus 1 equals a t minus 1. These kind of stochastic process is are what we called as Markov Chains ok.

So, typically the values that this random variable takes; so X is a discrete random variable which means it can take say accountably many values. So, let us look at this equation a little more carefully.

(Refer Slide Time: 05:52)



So, the random variable X t can take accountably many values, let us say the values that it takes we will call it as p 0 p 1 p n and so on. It could take infinitely many values, but accountably infinite. And X t minus 1 could have taken values q 0 q 1 q n and so on.

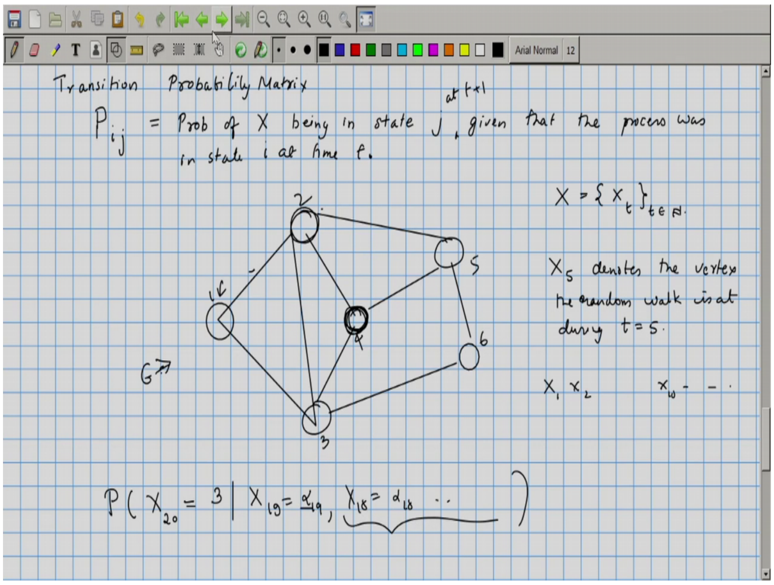
Now the entire evolution of these random variables; so, X t the value of X t is equal to let us say P i; so it takes one of these values. So, the probability that this is equal to P I, given the entire history is said for Markov processes or Markov Chains this is going to be equal to probability that X t is equal to P i given X t minus 1 equals let us say p j ok. So, let us just try and capture this by means of a matrix ok.

So these are the, on the rows there are these values p 0 p 1 P i P i plus 1 and so on could be an infinite matrix. And q 0 q its a q j so on are the values that X t minus 1 can take ok. So, what is a probability that X t is equal to p i, given X t minus 1 is equal to let say q j ok. So, we could just; so this is going to be this probability let us say if we call this as let us say x ij, ok. The probability that the random variable takes the value P i given that in the previous step or the previous random variable X t minus 1s value was q j. Let us denote that probability by x ij. So, this matrix in some sense captures the information corresponding to how these random variables transition as time progresses, ok.

So, now this matrix a priory could be dependent on t. If it is not dependent on t then those kind of stochastic process is we will call them as a Homogeneous Markov Chain ok. So, this matrix here captures the information corresponding to what is the probability that at the next stage the Markov process will be at a state P i given that at the previous step it was at q j, ok. And if this matrix has no dependence on t, or in other words for all t's if this matrix was the same matrix those kind of Markov process is or what we call as time homogenous Markov processes, ok.

And in this lecture and in this course we will bother only about time homogenous Markov Chains. So, time homogenous Markov Chains they can essentially be represented by a matrix we will call that as a Transition Probability Matrix.

(Refer Slide Time: 09:54)



So, the ij-th entry of this matrix will denote the probability of X or a random process being in state j given that the process was in state i let set time t. This is a time t plus 1. So, if at time t the Markov Chain is in state i at time t plus 1 what is a probability that the chain will be at state t plus 1. By the state of the chain we mean the value that the random variable X i takes ok.

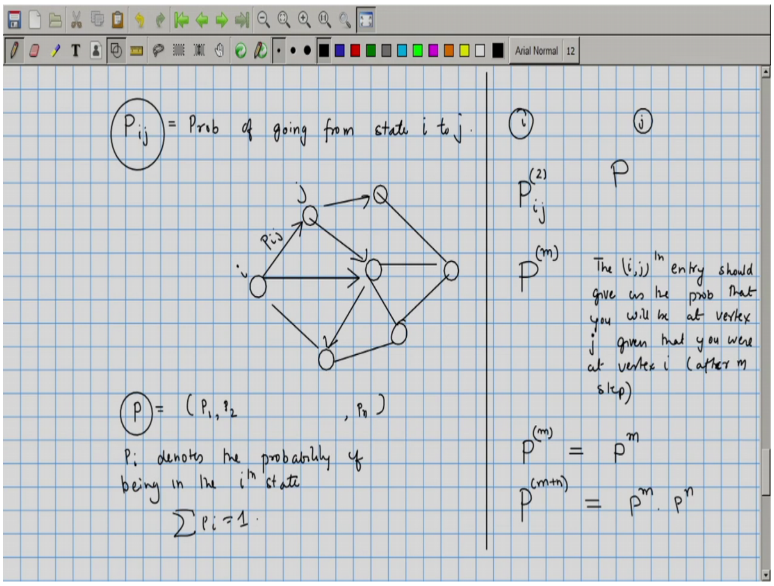
So, let see an example of a Markov Chain. So, we can imagine a random walk on a graph ok. So, let us imagine this graph ok. These are the states 1, 2, 3, 4, 5, 6. So, there are 6 possible states that the random walk could be in. So, the random walk or the stochastic process X will be all consisting of all these random variables X t that t belongs to natural numbers ok. So, X 5 for example; X 5 denotes the vertex the random walk is at during t equals 5 ok.

So, let us say you begin at this place at state 1 at time t equals 0. And then there are two outgoing edges toss a coin and go to either state 2 or depending on toss go to either 2 or 3. If you had gone to state 2 then at the next instance toss a coin and based on the toss two coins and based on whatever or of the outcomes go to 1, 3, 4, or 5 and so on. So, keep on walking on this particular graph, at any when you are at any particular vertex look at the out degree, and choose one of those vertices uniformly at random and move to that particular vertex and keep on doing this. And you keep on doing this can be used to define a sequence of random variables X 1 X 2 X 10 and so on ok. So, X 10 in particular will denote which is the vertex you are at time t equals 10.

So, this is a Markov Chain. And this is a Markov Chain because it does not really matter. So, the probability that to say X 20 is equal to 3 given X 19 is equal to let say something alpha 19, X 18 is equal to alpha 18 and so on. This probability does not really depend on the history by much, it only depends on which is a current state you are in. When you have reached let us say state 4 from here the transition is to one of states 2, 3, and 5. And that is not going to be dependent upon how you reached state 4.

So that probability, this transition probability depends only on the number of neighbours it has it is no dependence on how you reached 4 ok. So, this entire thing, this entire random walk can be represented using a matrix.

(Refer Slide Time: 14:40)



So, matrix this is a transition probability matrix, P ij is we will just write it as the probability of going from state i to j. Now, let us look at this a little more carefully. We can ask lot of questions when we have a Markov Chain like that, we will think about only discrete Markov Chains and we will assume that the number of states are finite ok. So, when the number of states are finite we can represent each of these states by a vertex of a graph and the edges denote whether you can transition from that particular state to some other state.

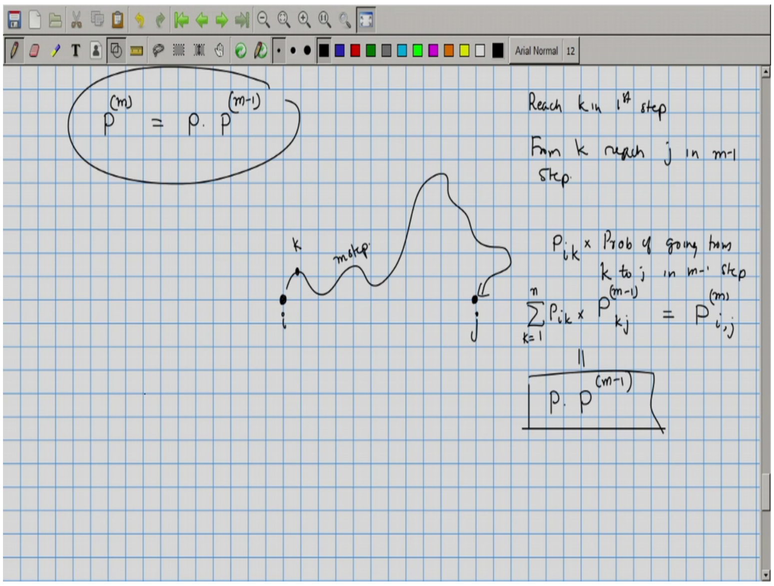
And on these edges you can put this P ij. So, if you are at state i P ij denotes a probability of going from state i to state j. So, the random walk on or graphs we can think of that as a generic Markov Chain. Now, you could ask this question: suppose we choose state at random and continue walking for some fixed number of steps say 100 steps. We could be at a variety of places, we could be at any of the vertices in this graph after 100 steps what are the probabilities that you are at some particular state.

So, we will denote that information by a probability vector. So, p is this vector where the components p 1 p 2 p n denotes P i denotes the probability of being in the i-th state. So, clearly summation over i P i is going to be equal to 1 ok. So, here when this we are abusing notation a little bit, this is a same matrix, this is we are using similar letter p for the matrix and we are using small p for the vector. But here when we write P ij there are two indices when we write P i just has one vertex when has one index.

Now, suppose instead of moving one step at a time if we when we want to compute the two step probabilities. If you are at step i, I mean if you are at vertex i what is a probability that you will reach vertex j in two steps ok. So, P ij denotes a probability that you will go from i to j in one step. This, if you are already at j i what is a probability that you will land up in j in the very next step. Let us denote by P ij two the probability of going from i to j in two steps. In general if we write p m, this denotes a matrix whose ij-th entry; so the ij-th entry of this matrix should give us the probability that you will be at step, you will be at vertex j given that you were at vertex i. So, this is after m steps.

So, imagine that you are already at vertex i. In m steps after m steps when you look at the Markov Chain what is a probability that you will be at state j, ok. Compute this for every i comma j and the matrix that you get by placing these ij’s, these probabilities is our matrix P m ok. So, what should P m be in terms of P? P is a matrix. So, ij-th entry gives us a probability of transitioning from state i to state j. So, P m will be equal to P raise to m. In fact, P m plus n will be equal to P raise to m times P raise to n ok. Why is this so?

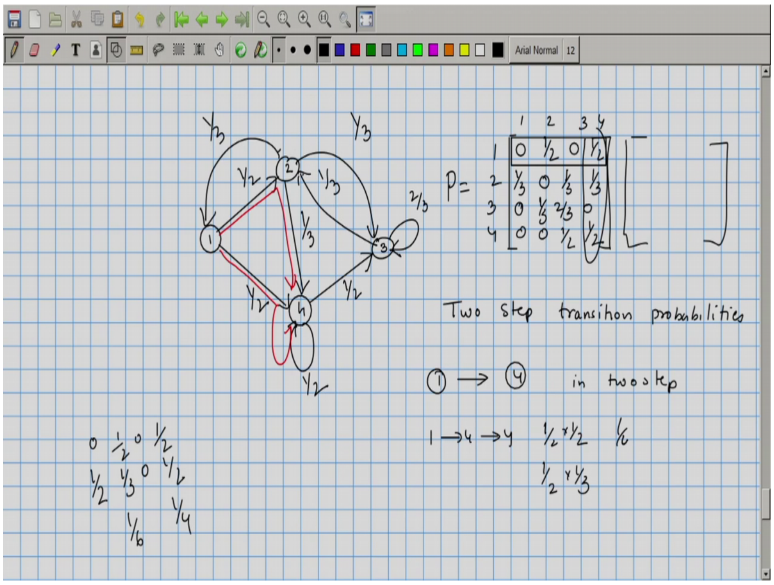
(Refer Slide Time: 20:24)



So, we will just verify that P raise to P m this is a matrix which were defined earlier, this will be equal to P times P raise to m minus 1 ok. So, let us say we were at state i and we need to reach state j in m steps ok. So, we could do this by choosing our first step, we will go to some particular vertex k and then from k we have to reach j. So, reach k in first step from k reach j in m minus 1 steps, and this k could be any vertex or any state in the Markov Chain.

So, the probability of this is; so for a particular k this probability is going to be P ik times probability of going from k to j in m minus 1 steps. But that is going to be equal to P ik times. If you look at the matrix P raise to m minus 1 inside that the k jth entry essentially gives us this. And this is for one particular choice of k, the total probability is going to be some over all values of k. So, k equals say if you had n states k equals 1 to n this summation will give us the probability. So, this will be equal to P m ij ok. And this is just P times P raise to m minus 1 in matrix form ok. So, this is going to b equal to; so this is what we wanted to show. So, let us see some worked out examples.

(Refer Slide Time: 22:57)



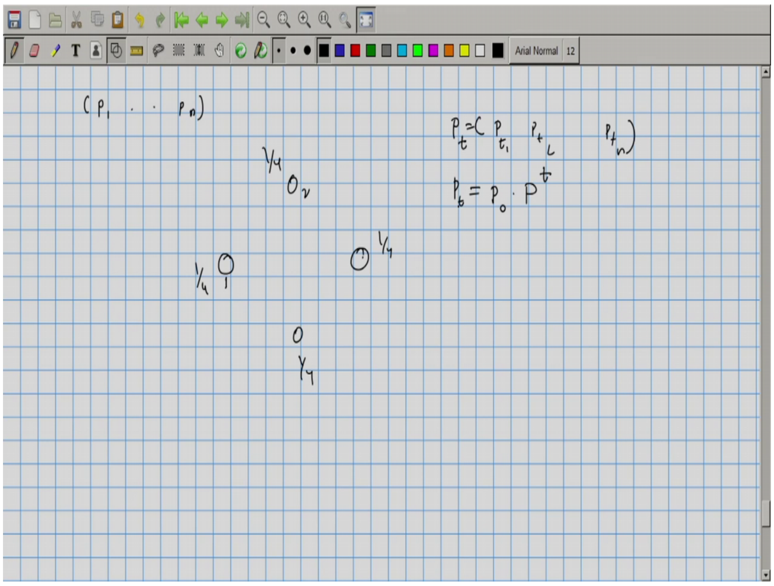
So, let us say we had Markov Chain with 4 states. This happens with half probability and you go here with half probability, from here let us say one-third. You go back here with one-third go here with one-third probability. And from this let us say you just go here and here this happens with probability half, this happens with half say this happens with one-third and this with probability two-third ok.

So, the transition probability matrix will be as follows: 1, 2, 3, 4 are the states 1 to 1 0, 2 to 2 is 0, 3 to 3 is 0 and sorry 3 to 3 is two-third and 4 to 4 is half. 2 to 1 is one-third and all these were one-third; 1 to 2 is half, 1 to 4 is half other is 0 3 to 1 is 0, 3 to 2 is one-third and this is 0; and 4 to 1 is 0, 4 to 4, 4 to 3 is half ok. So, this is our transition probability matrix for this particular Markov Chain.

And if you want to look at two step transition probabilities. Then, so let us just look at one particular example, if you wanted to go from 1 to let us say 4 in two steps. You could either go from 1 to 4 and then stay at 4, that is one way; you go here and then stay here. The other way would have been go here and then go. So, these are the only two possibilities and that will happen with probability.

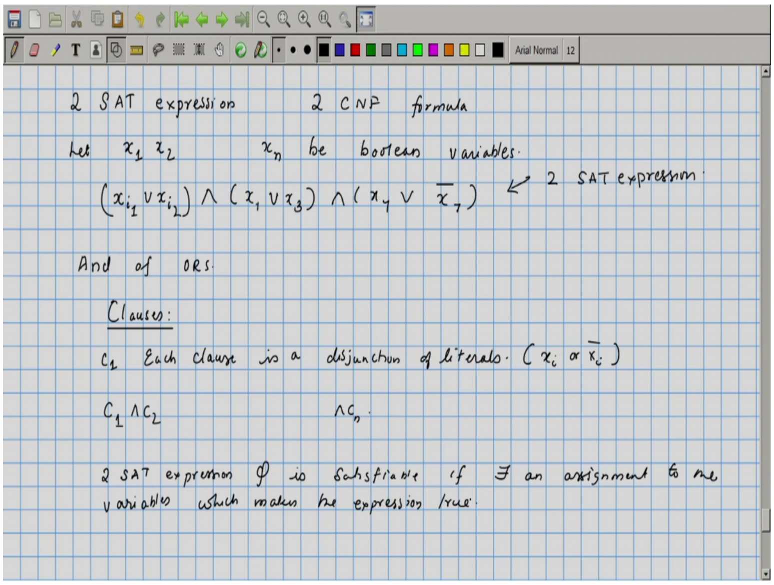
So, this happens with probability 1 by 2 into 1 by 2 and the other one happens with probability 1 by 2 times 1 by 3. If you had multiplied p with itself ok, the one four entry would essentially be 0 half 0 half multiplied with the 4th column ok. So, this will be multiplied with half 1 by 3 0 and half. So, that will give us one-sixth plus one-fourth. That is the same here ok.

(Refer Slide Time: 26:48)



Now another thing is: if we start at one of the states say p 1, so this way were are initial probability distribution on the states. So, each state let us say we were in state 1 with probability 1 by 4, state 2 with 1 by 4, state 3 with 1 by 4, and state 4 with one all these 4 states were equally likely. Then at the next stage the probability that if you look at if you indicate by p t the probability p t 1 p t 2 p t n; let us say this denotes the probability that you are in state i at time t. Then this is going to be equal to p 0 times P raise to t ok. This also you can easily verify this. Now, that we know something about Markov Chains. We will now discuss a problem which we can solve using these ideas.

(Refer Slide Time: 28:02)



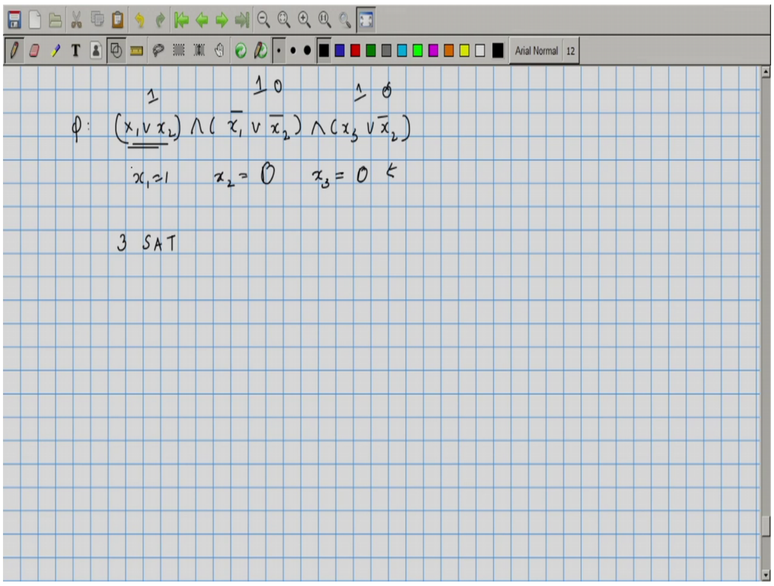
So let me introduce what is called as a 2-SAT expression. So, let x 1 x 2 x n be Boolean random variables or sorry Boolean variables ok. Now you can combine these using disjunction and conjunction; and we will combine it in the following way. So, let us x i 1 or x i 2 and so let say x; so some something x 1 or x 3 and x 4 or x 7 complemented and so on ok. So, exponential of this kind will be called as a 2-SAT expression. Formally this will be and of ORS or conjunction of disjunctions. So, this is also called as a 2 CNF formula ok.

So, no firmly we have clauses and each clause is a disjunction of literals, where are literal is x i or x i bar; x i bar denotes the complement of it ok. So, you have an expression like this, and what we have is the. So, if C 1 is a clause we have an expression of the firms C 1 and C 2 and C n ok

Now, what we are interested in is; so this can be taught of as a logical formula, and if you fix the values for x 1 x 2 x 3 up to x n this expression will have a true or false value you can evaluate the expression ok. So, a 2-SAT expression can be evaluated we call a 2-SAT x expression satisfiable. So, let us say phi is satisfiable if there exists an assignment to the variables which makes the expression true.

So, if we evaluate this particular Boolean expression after plugging in the values the variable if you get 1 or if you get true, then you will say that that particular expression is satisfiable. So, our problem is we are given a 2-SAT expression and we want to know whether the expression is satisfiable or not.

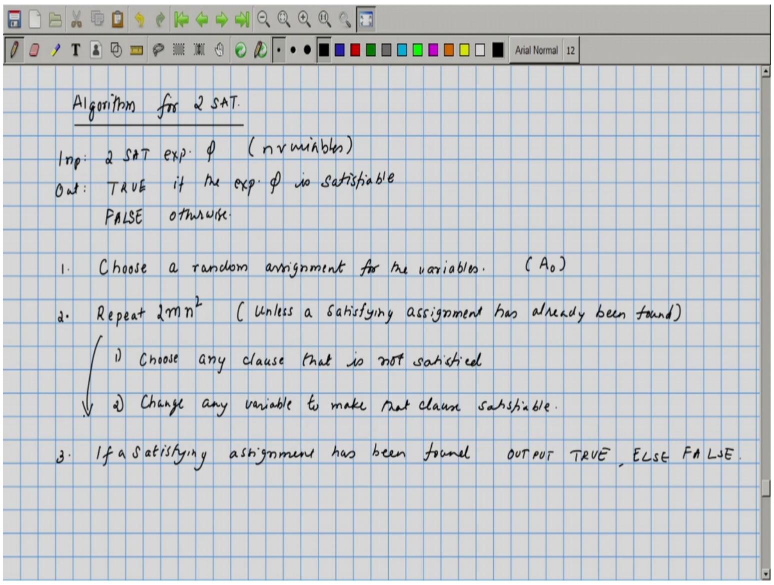
(Refer Slide Time: 31:41)



Suppose, this was our expression; if we put x 1 equals 1, x 2 equals 0 and x 3 equals 1 or x 3 equals 0 you can see that this expression will evaluate to. So, this will evaluate to 1, this will evaluate to 1; because x 2 bar is 1 and this will also evaluate to 1 ok. Whereas, if x 2 was 1, and this will evaluate to 1 this expression will evaluate to 0 and this expression will evaluate to. So, this will evaluate to 0 and this will evaluate to 0 ok.

So, what we are interested in is given a Boolean expression a 2-SAT Boolean expression we want to know whether there is a satisfying assignment. This is an unsatisfying assignment, whereas if you had taken x 1 equals 1 x 2 equals 0 that is a satisfying assignment. So, does there exists a satisfying assignment. There is a deterministic polynomial type algorithm to solve this, but we will just see a randomized algorithm. And we will convert, we will try and build randomized algorithm for even 3-SAT, where instead of two literals we will be having three; we will look at clauses which has three literals. So, let us see what the algorithm is.

(Refer Slide Time: 33:43)



So, the input is a 2-SAT expression, and the output is true if the expression phi is satisfiable and false otherwise ok. So, what are the steps of this algorithm? So, what we will do is very simple, we will just take a random clause; initially we will start with a random assignment all the vertices or all the variables will be given will be assigned true or false randomly. And then after that what we will do is we will look at these clauses, if any of the clauses is unsatisfied you can see that all the clauses need to be satisfied, because it is a conjunction, so And of ORS. So, only if every clause is satisfied the formula will be satisfied.

So, we will pick one of the clauses which is unsatisfiable for a given assignment, and then randomly change one of the literals in that to make that particular clause satisfiable. And then again reach if the entire expression is satisfiable we will keep on repeating it for some number of times ok. When we keep on repeating it some number of times, we will say that if luck is on our side we will get a satisfying assignment ok. In fact, what we will show is for every expression which is satisfiable luck will be on our side. In other words with a very high probability we will come up with the, we will find out a satisfying assignment.

So, we will begin as follows: choose a random assignment for the variables. So, let us call this as the initial assignment or A 0 and then repeat the following steps ok. So, while repeating we will check unless a satisfying assignment has already been found. So, in between if you find a satisfying assignment at any stage you will quit, otherwise you will repeat these steps. The number of times, so we will write it as m times say two times m times n square, where n is the number of variables. Choose a random clause or choose any clause does not have to be random choose any clause which is that is not satisfied. Change any variable to make that clause ok. And then if a satisfying assignment has been found output TRUE else FALSE; so this is our algorithm.

We will start by choosing a random assignment. After choosing a random assignment we will evaluate the expression on this particular assignment. If the expression evaluates to true then we will say that will output that the expression is satisfiable. If even one of the clauses is unsatisfied we will pick one of those unsatisfied clauses and change a variable inside that, and see if we get a satisfying assignment. We will keep on doing this process for m 2 times m times n square; m is some parameter that we can fix later on.

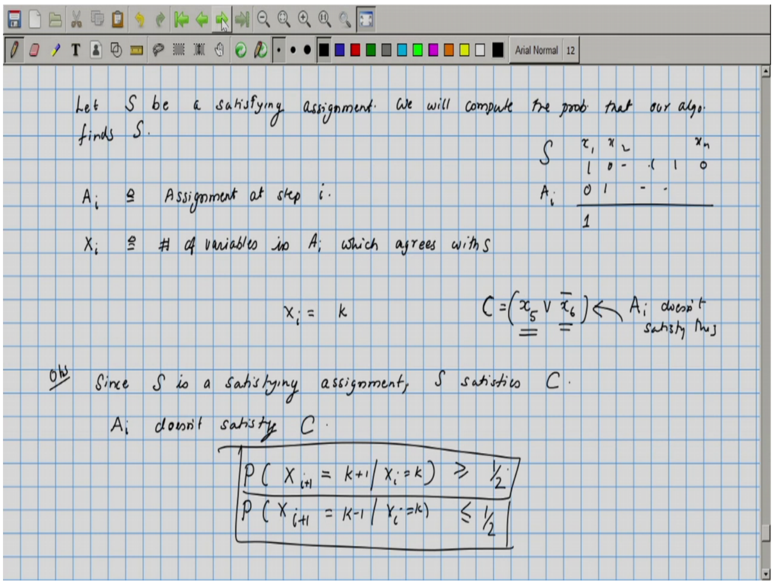
The higher the value of m, the higher the probability of success would be the. So, success here would mean that if the given expression or the 2-SAT expression was the 2 C n of expression was satisfiable; then we would have to say satisfiable and if it is not satisfiable we should say not satisfiable or false. It is clear, that this algorithm whenever it is presented with an input which is not satisfiable it is never going to find the satisfying assignment. And therefore, it is always going to give the correct answer which is false ok.

It will output false as the answer and that is the correct answer. But when the expression is satisfiable may be there is a minor possibility, there is a small probability that we will never find out that satisfying assignment when it will find the probability of such events ok.

So, how do we analyse this algorithm. So, we will look at the following we will analyse it in the following way. So, imagine any satisfying assignment, we need to bother only about CNF formula or 2-CNF formulas which were satisfiable, because unsatisfiable formulas of course we are not going to cause we are going not going to make a mistake on that.

So, if you have a satisfying formula, it could have many satisfying assignments, but we will compute the probability that it will find a particular satisfying assignment. And we will show the that probability itself its reasonably high ok.

(Refer Slide Time: 40:52)



So, let S be a satisfying assignment, we will compute the probability that our algorithm finds S. Of course, the algorithm may not find S, it might find some other satisfying assignment and quit ok. But this probability certainly is going to be, when the probability that it will find this particular S if we compute that probability if that probability is high that is good enough for us ok.

So, we will introduce certain random variables. So, let us say A i will denote the assignment of the partial assignment at step i. So, here the algorithm keeps on modifying the random assignment and at the i-th step there is a certain number of; I mean there is a certain assignment of values to the random to the variables of our expression. So, A i denotes that assignment. And let X i will denote the number of variables in A i which agrees with S. So, S let us say that is the following assignment x 1 x 2 x n; 1 0 1 1 some binary string denoting the assignments to the variables x 1 to x n. And A i is the assignment at some stage this is 0 1 something.

So, look at the number of; so look at each variable and if those variables in the assignment that is the current assignment in the algorithm if it matches with S. Then we will add 1, then our so if the i-th variable matches you can add one otherwise 0 and you summit up over all. That sum at the i-th stage number of variables which agrees with S at stage i at step i is denoted by X i.

So, now we want to compute certain probabilities. If you look at this random variable X i, X i initially is some number let us say p or let say alpha and then it keeps on increasing. It might even decrease at a certain stage, but we want to know what is a probability of it increasing. Now if x. So, x 1, if you look at x 1 x 2 x we will go on till mn’s two m n square steps. So, this is a sequence. So, let us imagine that instead of two m n square steps we keep on going forever ok. We never stop, we stop only when we get a satisfying assignment ok; now or when we get this particular assignment S.

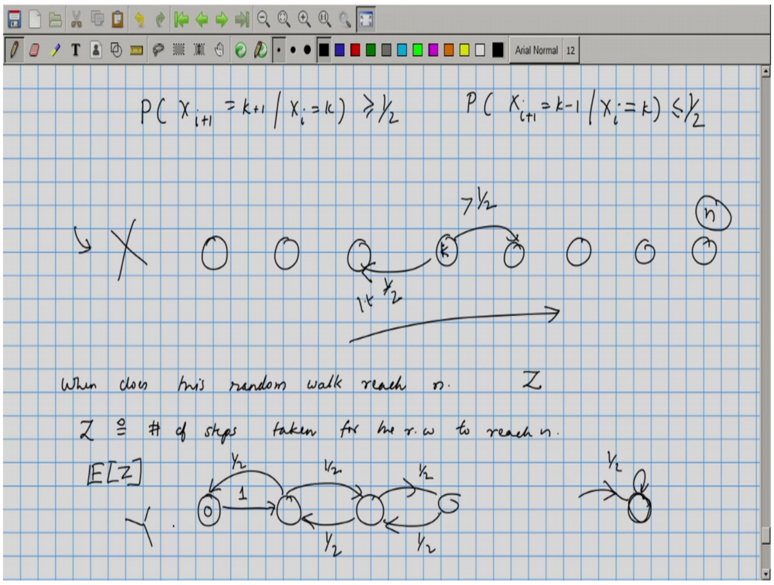
Now at any stage if X i is equal to let us say k at the next stage what we do is; we had picked one random clause or one clause which is not satisfied by the current assignment and changed its value when change the value. So, let us say we had picked clause x 5 or x 6 bar, so this is the clause that we picked at some stage. This is not satisfied by A i. So, A i does not satisfy this, and we flip the value of one of these x 5 or x 6 we choose randomly ok; so this any variable to make the clause satisfiable, ok.

Now, since S is a; so this is an important point. This is the observation. Since S is a satisfying assignment ok, if we call this clause as C, S satisfies C, ok. But the assignment that we have is A i and A i does not satisfy C, because C is one of those clauses which A i which is currently unsatisfied. So, since A i does not satisfies C, at the next stage X i from, so X is value can go from k to k plus. So, X i plus 1; so X i plus 1 can be equal to k plus 1 with probability greater than or equal to half ok. If we randomly choose one of the variables in this clause and flip it X i is going to increase to k plus 1 with probability half.

So, now we can compute this probability. Probability that X i is equal to k plus 1 given sorry; X i plus 1 equals k plus 1 given X i equals k ok. Since we had taken an unsatisfied clause and flipped one of the variables in that clause ok. So, one of these surely does not agree with S, ok. It could be the case that both of them does not agree with S, but at least one of them does not agree with S. So, if we choose that randomly we have at least half probability that in the resulting new assignments the number of variables that agree with S is one more than whatever it was in the previous assignment. And therefore, we can write the probability that X i plus 1 is equal to k i minus 1 given X i equals k is going to be less than or equal to half, ok.

Since the probability of this is this probability is greater than half the other probability is going to be less than half ok. So, does this behave like a Markov Chain; will not really, if this was equal then it would have been a Markov Chain. But we can convert this into we can construct another Markov Chain which behaves nicely.

(Refer Slide Time: 48:56)



So, what we have here is: probabilities that X i plus 1 is equal to k plus 1, given X i equals k is greater than or equal to half, and probability that X i plus 1 is equal to k minus 1 given X i equals k is less than half ok. So, we can think of this as a random process not a Markov process, but a random process or random walk. So, from each step if you are at step i or let us say if the value is k it goes to k plus 1 with probability half or more than half and it goes back with probability less than half ok: so less than half greater than half.

So, this particular random process has a tendency to move forward. What we are interested in is a following thing: X n. So, X when does is if you look at this random process when does the random process hit the value n: when does this random walk reach n?

So, we need to; so if we call that as let us say a random variable Z, we need to the Z denotes the number of steps taken for the random walk to reach n, ok. We need to compute expectation of Z. If expectation of Z is reasonable number if it is small; that means, we will quickly find a satisfying assignment. When Z is equal to when the random variable when the random walk reaches this state n it means that we have found an assignment which agrees with S, and S being a satisfying assignment we have found a satisfying assignment ok.

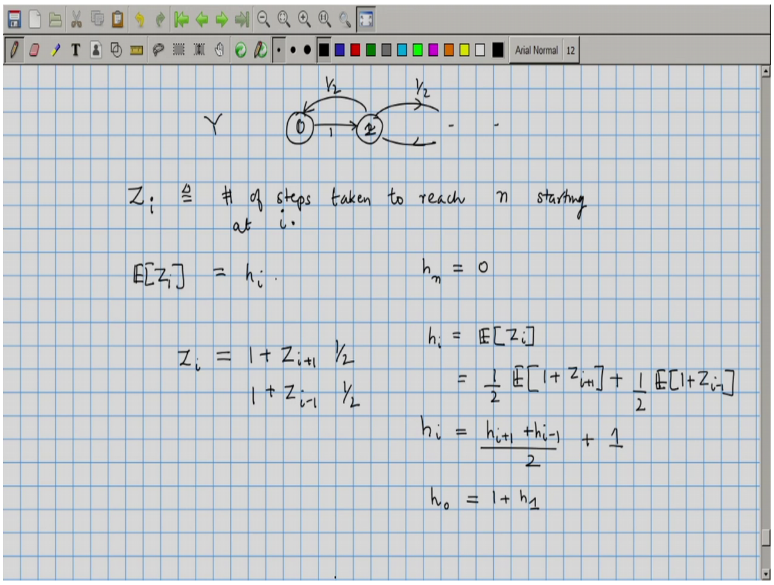
So, how do we compute this? What we will do is instead of looking at this we will look at the simpler random process wherein, from step 0 you go forward with probability 1 and from everything else go forward or backward with probability half each; so now this is clearly a Markov Chain, ok. May be from the final state you can just stay there ok. So, there is no returning from the final state.

So, if you look at this particular random walk, this random walk is surely slower than the; so, let us call this as the random walk Y and this is a random walk X. Y is surely slower than X. Slower in the sense X has a tendency to move forward, because it takes the forward edge or it goes forward from state k to k plus 1 with probability greater than half and comes back with probability less than half; less than or equal to half.

So, now if you think of moving another parallel random walk such that the probabilities are half for both forward edge and the backward edge then clearly the random walk Y is expected to be slower. There could be instances where Y is faster than X, but in an expected sense X is going to always have a smaller expect. The number of steps taken for X to reach; let us say any particular state n is going to be smaller than the number of steps taken for Y to reach the same step.

So, the expected number of steps taken by the random walk Y to reach the state n will be a bound an upper bound on the number of steps on the expected number of steps taken by the random process Y; sorry random process X. X being non-Markov the analysis of the expectation is difficult, whereas Y by virtue of being a Markov process we can compute this expectation quickly.

(Refer Slide Time: 53:56)



So, let us just look at that particular computation. We will write this down as follows. So, we have a random walk from 1 you go forward with probability 1, from 2 you come back half go forward half, and you continue with this ok. So, this is the random process Y. And let us say Z i will be the random variable. So, this denotes number of steps taken to reach n starting at 1 or let us call this as 0, so starting at i.

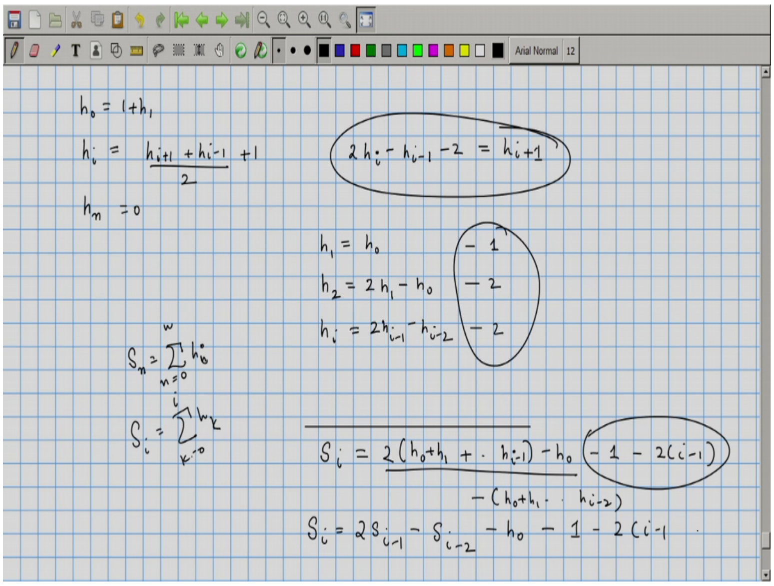
So, if you start from the state i, how much time does it take to reach step n. This could this is a random variable that random variable is what we call as Z i. And the expectation of Z i we will denote it by. So, this we will call it to be h i ok. So, h i is defined to be the expected number of steps required to reach state n starting from state i ok. So, h 0 is the quantity that is of importance to us in the algorithms analysis, if we show that h 0 is less than some particular quantity then our algorithm. I mean if it if we, show that this is less than n square then that means our algorithm has a chance to find the, when if you run the algorithm for n square steps we are going to we expect to find a satisfying assignment, ok.

So, let us first compute the expectation of Z i ok. So, we can write this in term by means of recurrence relations. So, h n is equal to 0, because if you start at the n-th step you immediately reach the n step you do not require any additional step. Now h i is going to be by definition this is a expectation of Z i this is equal to at Z i, you can with probability half i mean if you were at if you start your random walk at the i-th step with probability half you can go forward and with probability half you can come back.

So, Z i could be equal to 1 plus Z i plus 1 with probability half and 1 plus Z i minus 1 with probability half ok. Z i denotes the number of steps taken from the state i. If you had gone forward then the total number of steps taken would be 1 plus Z i n that happens with probability half. And if you had gone back you would take Z i minus 1 steps and that would happen with probability half. So, the expectation of Z i is half into expectation of 1 plus Z i plus 1 plus half into expectation of 1 plus Z i minus 1 ok.

So, this will be equal to h i plus 1 plus h i minus 1 by 2 plus 1 ok. So, this is a recurrence Z we can write. And h 0 is certainly going to be 1 plus h 1 because if you are at state 0 you straight away go to state 1. So, these are the recurrences. Now we need to solve this recurrences and evaluate what is h 0.

(Refer Slide Time: 58:43)

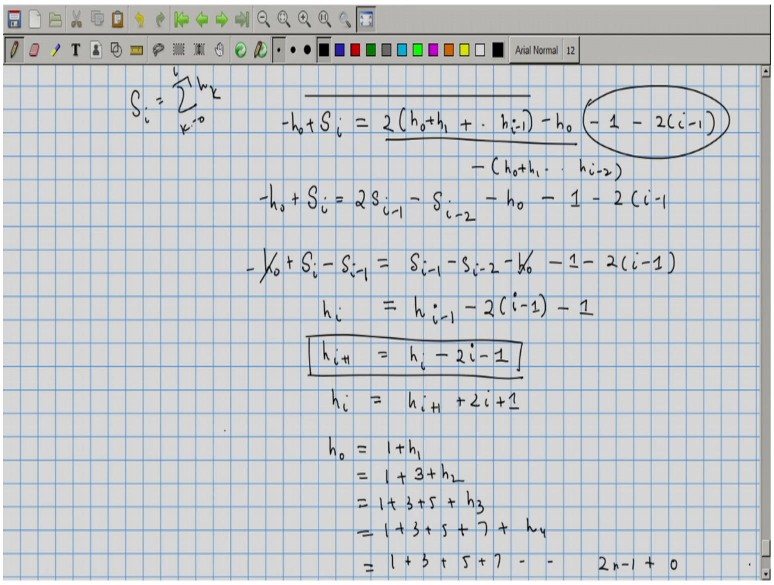


So, this is what we will work with; h 0 is equal to 1 plus h 1 h i is equal to h i plus 1 plus h i minus 1 by 2 plus 1 and h n is equal to 0. So, this can be rewritten as h 2 h i minus h i minus 1 minus 2 is equal to h i plus 1. So, if we use this recurrence we can write this entire system of recurrences as: h 1 is equal to h 0 minus 1, h 2 is equal to 2 times h 1 minus h 0 minus 2, h i is equal to 2 times h i minus 1 minus h i minus 2 minus 2, and h n is equal to 0.

If you add these things the LHS will add up to summation h n. So, let us denote S n is equal to summation h n, n going from sorry h i, i going from 0 to n ok. So, this is S n this is equal to 2 times h 0 plus h 1 plus h n minus let say we will sum this up till; we will just sum this up till h i. So, this will be S i. So, S i is summation k equals 1 to 0 to i h k. So, this is S i this is 2 times h 0 plus h 1 to h i minus h 0 minus 1 minus 2 times i minus 1 ok.

So, this we can write it as oh sorry there is an additional term that we have missed. So, these two these sum to this much, and the other terms sum to minus h 0 plus h 1 to h i minus 2 ok. So, we can write this as S i is equal to 2 times S i minus 1 minus S i minus 2 minus h 0 minus 1 minus 2 i minus 1.

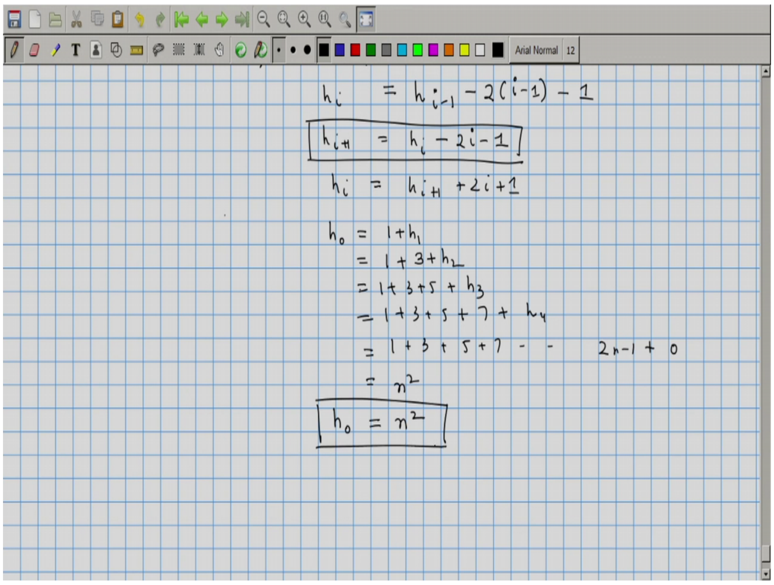
(Refer Slide Time: 61:57)



So we can take one of these i minus 1 to other side. So, we will get S i minus S i minus 1 is equal to S i minus 1 minus S i minus 2 ok. So, this summation is a minus h 0, because we did not add the first term is not there; so minus h 0 minus h 0 minus 1 minus two times i minus 1. So, S i minus S i minus 1 is going to be h i. So, this is h i these terms cancel is equal to h i minus 1; sum to i minus 1 minus sum to i minus 2 is h i minus 1 minus 2 times i minus 1 minus 1 ok.

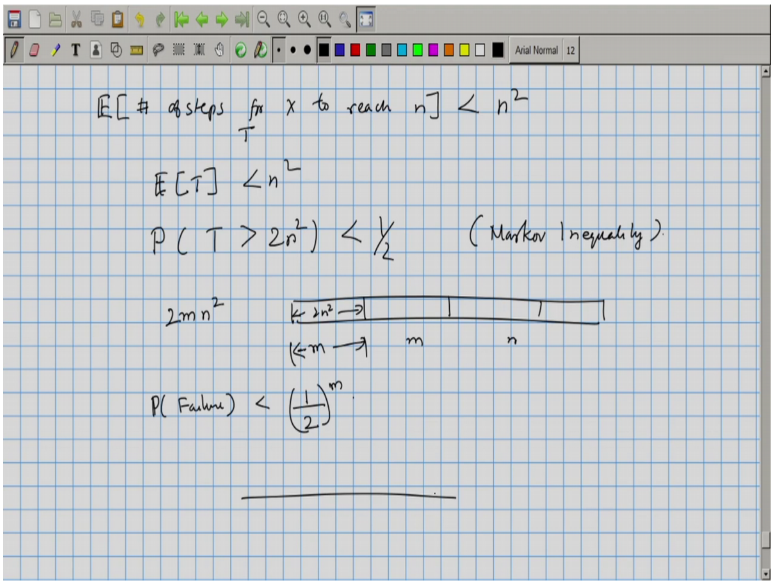
So, h i plus 1 we can be written as h i minus 2 i minus 1 ok. So, this is the expression that we will use ok. So, now what is h? So, h n is equal to 0 h 0 is what we are interested this is equal to 1 h 1 plus 1 this is equal to h 1 is by this expression two h 1 is equal to h 2 minus. So, in other words h i is equal to h i plus 1 plus 2 i plus 1. So, therefore, h 0 is equal to 1 plus h 1 and this is equal to 1 plus h 1 we can write it as 3. So, i here is 1, 3 plus h 2 and this is equal to 1 plus 3 plus 5 plus h 3 that is equal to 1 plus 3 plus 5 plus 7 plus h 4 and so on. So, this will give us 1 plus 3 plus 5 plus 7 plus up to 2 n minus 1 plus h n, and h n is 0, and this sum this is equal to n square.

(Refer Slide Time: 64:21)



So, we know that h we can now conclude that h 0 is equal to n square ok. So, what this means is: if we had taken the random walk X that would be going at a faster rate than Y; Y itself will hit the state n in n square times. So, X is expected time for X to reach the state n is going to be less than.

(Refer Slide Time: 65:02)



So, expected number of steps for X to reach n is going to be less than n square ok. So, if we denote this expected number of steps by the random variable T the expectation of T is going to be less than n square. And therefore, probability that T is greater than 2 n square is going to be less than half; this is a Markov inequality ok.

So, since T is greater than 2 n I mean the probability of T greater than 2 n square is less than half, if you are repeated for m n square 2 times m n square steps. And if we imagine all these steps I mean these two m n square steps to be broken down into m blocks each of size 2 n square, in each of these blocks you have less than half ;you have a less than half probability of not finding an assignment.

So, the overall probability: probability of failure is when all these m steps or m stages fail to find the satisfying assignment. That happens with probability less than 1 by 2 raise to m ok. That is an exponentially small probability.

So, that concludes the algorithm and its analysis for the algorithm and the so, that concludes the 2-SAT algorithm and its analysis.