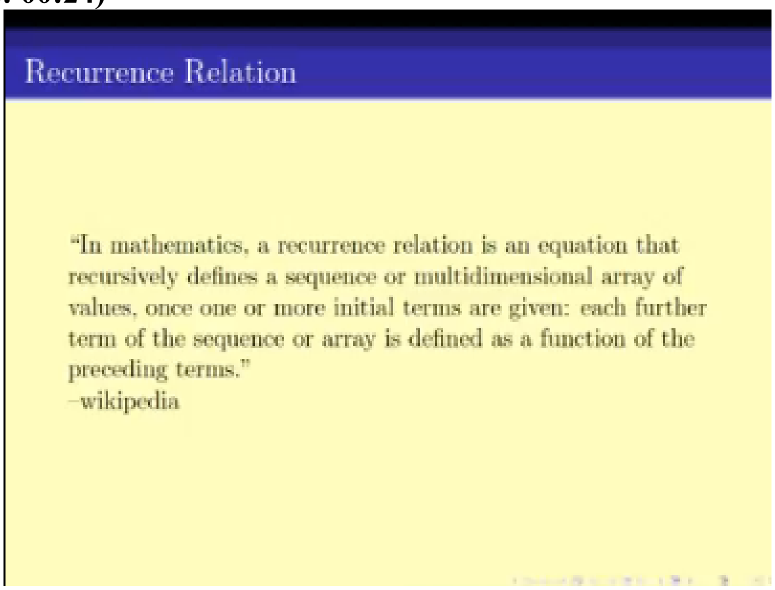
Discrete Mathematics
Prof. Sourav Chakraborty
Department of Mathematics
Indian Institute of Technology - Madras

Lecture - 38
Solving Recurrence Relations (Part 1)

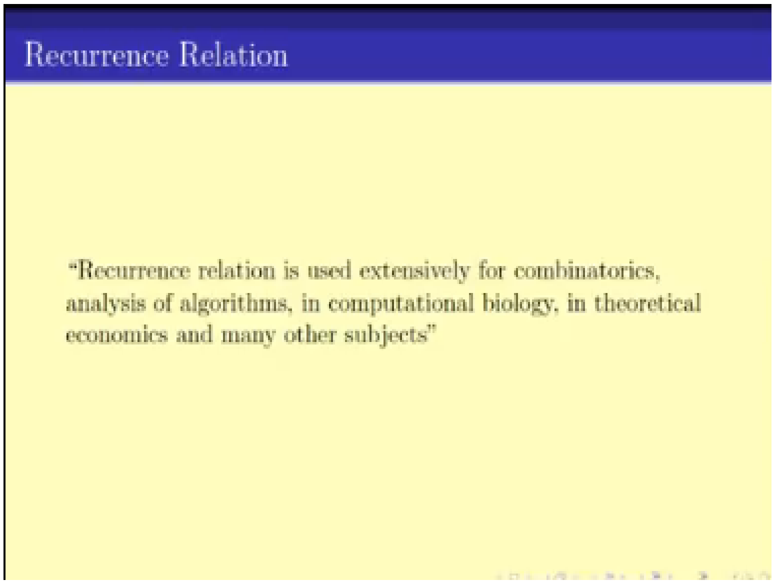
Welcome back. So in last few lectures, we have seen how to use the recurrence relations to model various counting problems.

(Refer Slide Time: 00:24)



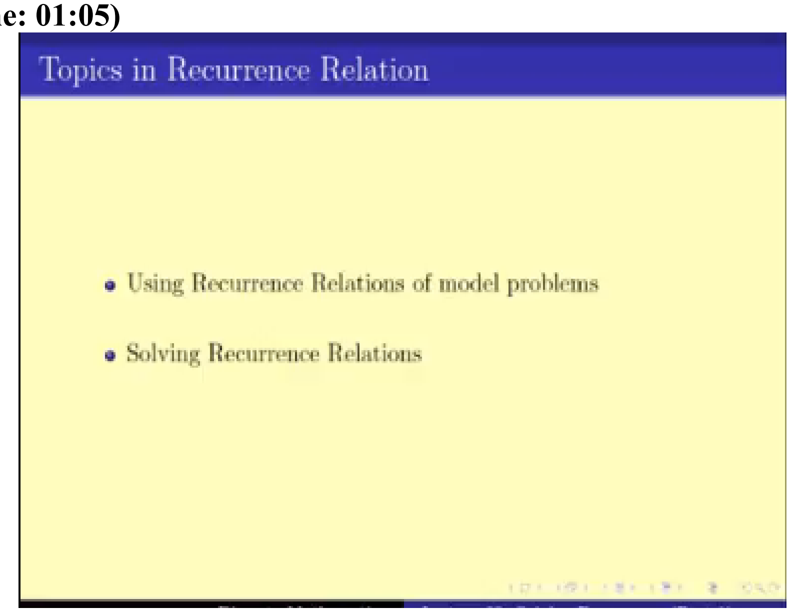
Now, as we have told recurrence relations are very essential part of mathematics or particularly in counting. So in fact, recurrence relation is an equation that recursively defines a sequence or multi-dimensional array of values where there are some initial terms and the n^{th} term is defined as a function of the preceding terms.

(Refer Slide Time: 00:47)



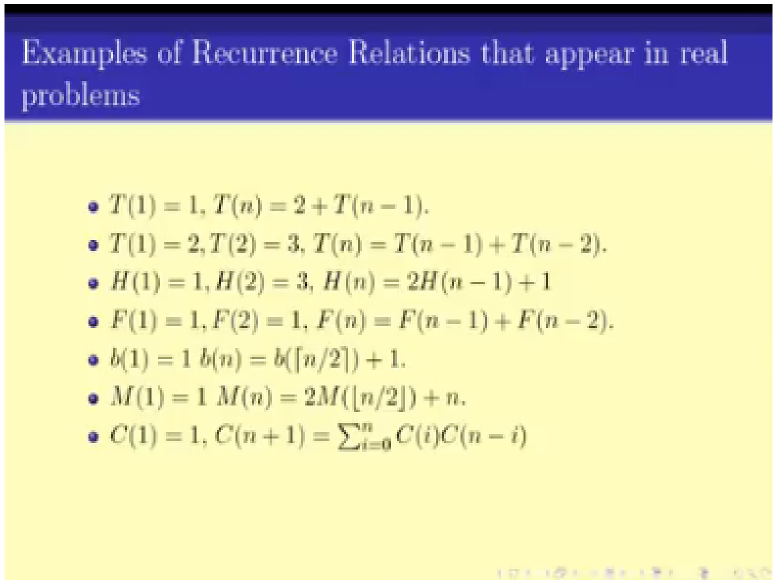
Recurrence relation is extensively used for combinatorics, analysis of algorithms, in computational biology, in theoretical economics and many other subjects. So we have already seen some of the recurrence relations.

(Refer Slide Time: 01:05)



And we have seen how recurrence relations can be used to model various problems, particularly combinatorics problems. But we have not seen how to solve the recurrence relations. In this video, we will be focusing on how to solve recurrence relations. So here, some of the recurrence relations that do appear in real life problems.

(Refer Slide Time: 01:36)

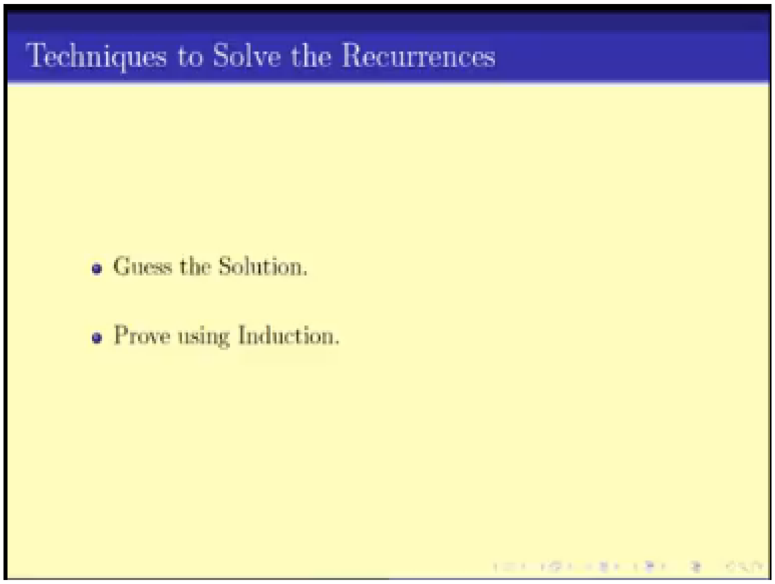


So the first one says, T(1) = 1 and T(n) = 2 + T(n - 1). The second one say, T(1) = 1, T(2) = 2 and T(n) = T(n - 1) + T(n - 2). Or this one is what we got from the Tower of Hanoi problem H(1) = 1, H(2) = 2, and H(n) = 2H(n - 1) + 1. This one is basically what we have the Fibonacci series where F(1) = 1, F(2) = 1 and F(n) = F(n - 1) + F(n - 2).

This is what is known as the Fibonacci series is quite a famous series that appears again and again in real life. Then this one b(1) = 1 and $b(n) = b(\lceil n/2 \rceil) + 1$. Then we have M(1) = 1 and M(n) = 2M [$\lceil n/2 \rceil$] + n. So these two appears in various algorithms, particularly, the binary search algorithm and the merge sort algorithm that are very popular in the algorithm's literature.

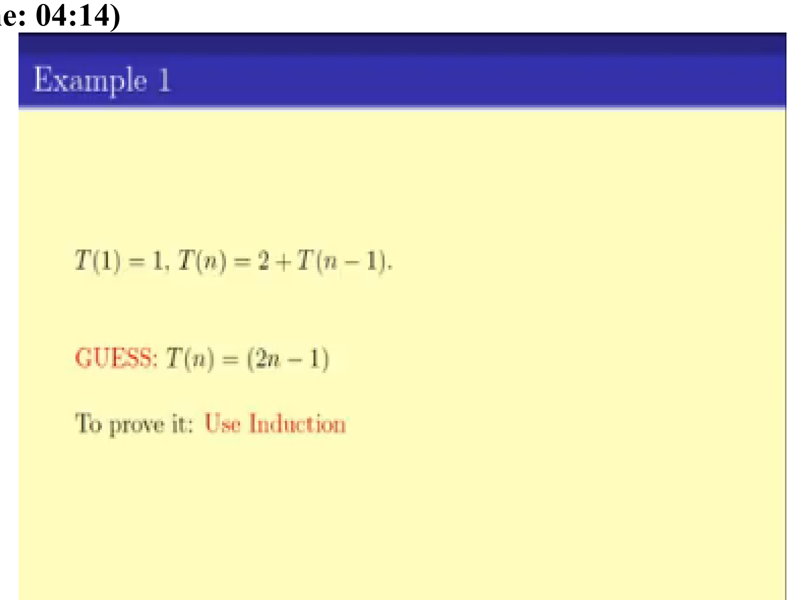
And then we have applied some of complicated one. C(1) = 1 and $C(n+1) = \sum_{i=0}^{n} C(i)C(n-i)$. Now for all of them, we have to now understand how one can solve them. So what is the technique for solving any of these recurrences? So, how to solve these recurrences?

(Refer Slide Time: 03:46)



Now, the first technique that we are going to look at is, the simple thing of guess the solution and prove using induction. In this video, we will see how this technique is useful. In next video, we will see how one can guess the solution. So say here we have this example.

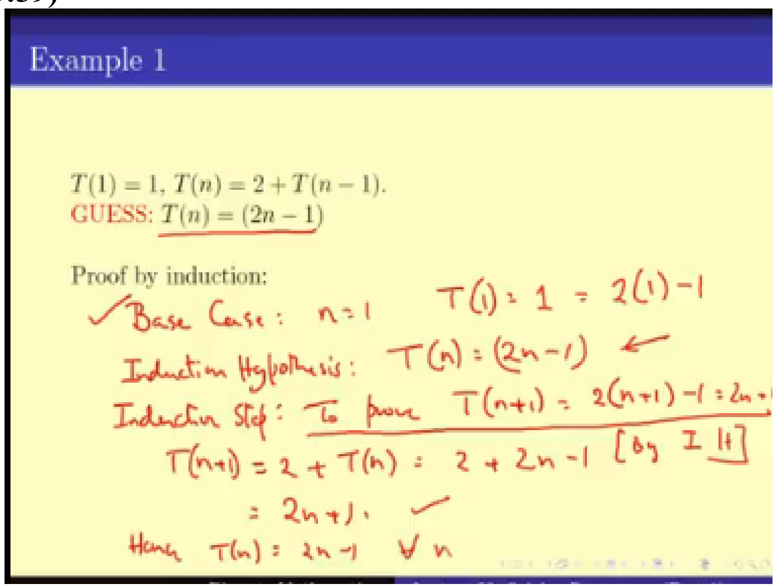
(Refer Slide Time: 04:14)



T(1) = 1, T(n) = 2 + T(n - 1). Now how do you solve this particular problem? Now first of all, if somehow magically you can guess this number, then we are great. For example, if I tell you, that guess where T(n) = (2n - 1). Now if this is the guess that we made, then we can try to prove the statement using induction. So the technique is first to guess and then prove by induction. Now I have skipped a big jump of how to guess this number.

We will see the technique of guessing in the next video as well as in the next whole week. Guessing the solution for the recurrence relation is possibly the most challenging part of solving the recurrence relation. But in this video, we will be focusing on how to solve the guess if we have the induction, if we have the guess right. If we guess the thing right how to prove it? So how do you prove by induction?

(Refer Slide Time: 05:39)

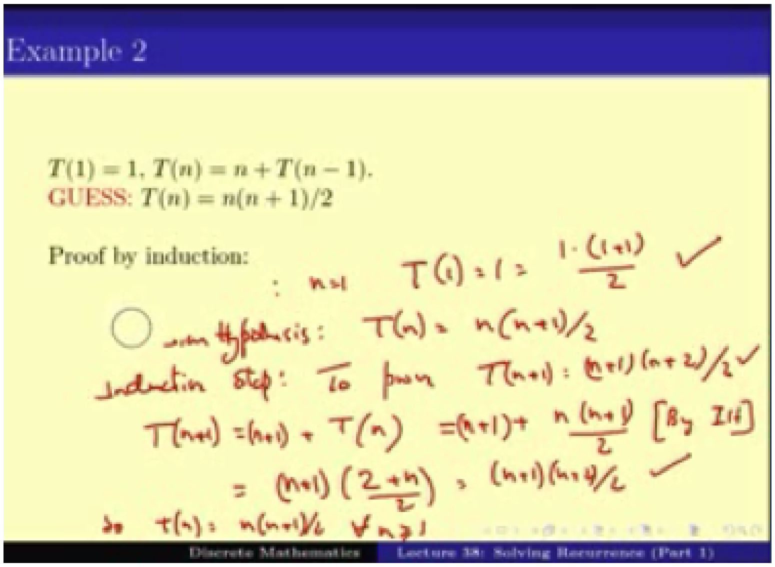


Now if you remember, so we should have a base case. In this case, base case is say n = 1 and we have T(1) = 1, this is a something that is given and which is T(1) = 2.1-1 = 1. So this value is correct for T(1). And now, we have the induction hypothesis, induction hypothesis which says for some n, T(n) = 2n - 1.

What is the inductive step? Inductive step is to prove the same statement T(n + 1) = 2(n + 1) - 1 = 2n + 1. How do you prove it? Now, of course by the thing that is given to us T(n + 1) = 2 + T(n) = 2 + 2n – 1 (*by induction hypothesis*) = 2n + 1 and that is what we had to prove. Hence, we are done.

Hence, we are proved the inductive step that means T(n) = 2n - 1 which is for all n ≥1. So this is the proof by induction once we have the guess right. So let us go over the next one.

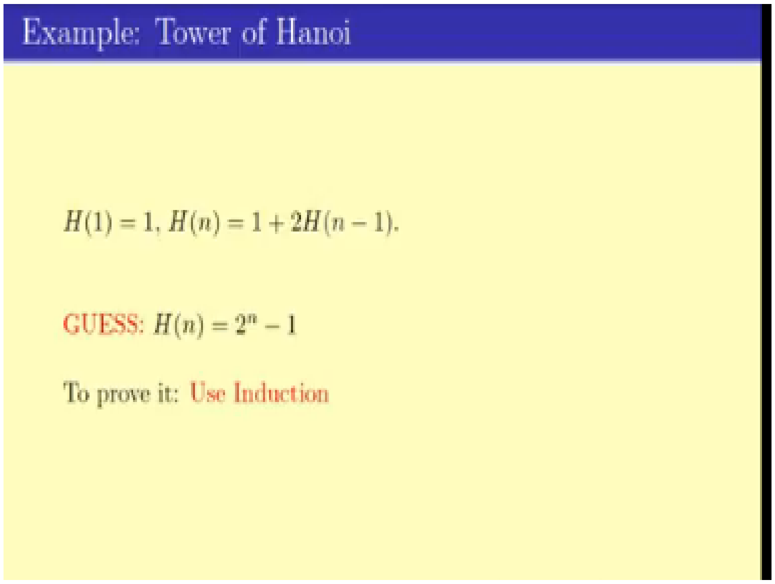
(Refer Slide Time: 08:22)



One more example $T(1) = 1$ and $T(n) = n + T(n - 1)$.

Again first of all, you have to guess it and let us imagine that somebody just comes up and manages to guess it correctly and say somebody comes and says that $T(n) = n(n+1)/2$.
.Now once someone has guessed it, we have to prove it, we have to ensure that the guess is right and to get that is true, we have to again use induction. So like in the earlier case, we have to again prove this one by induction and let us see how we prove it again.

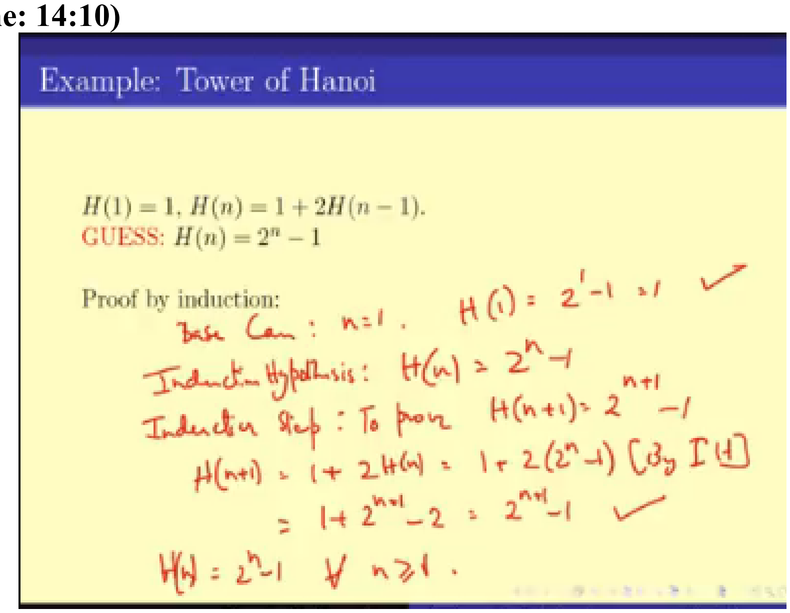
Say base case, $n = 1$, of course $T(1) = 1.(1+1)/2 = 1$.
Now we have the induction hypothesis, $T(n) = n(n+1)/2$.
Now in inductive step, we have to prove that $T(n + 1) = (n + 1). (n + 2)/2$. Now how do you prove it?
$T(n+1) = (n+1) + T(n) = (n+1) + n(n+1)/2$ *(by induction hypothesis)* $= (n+1)(2+n)/2 = (n+1)(n+2)/2$
which is what we had to prove.
So we have $T(n) = n(n+1)/2$ for all $n \geq 1$. Again the idea is simple if you can guess the value correctly for T(n), then you can prove what T(n) is by induction. Let us see one more example, what can be the various guesses?
(Refer Slide Time: 12:48)



So, this is we say Tower of Hanoi problem, right, so H(1) = 1 and H(n) = 1 + 2H(n - 1). Again, we first have to guess it. Now, what is the guess here? Say the guess is H(n) = 2^n - 1 and again we have to prove this one by induction. Note here that if you guess it wrong, we will not be able to prove it by induction or we are not be able to prove it.

So thus, only if you guess it right we will be able to prove this statement. So there are people who actually come up with these cases by some various intuitions of their brain but there are some techniques also which will help to come up with the correct guesses which we will study in next few lectures. But again for this particular problem, how do we prove this statement?

(Refer Slide Time: 14:10)

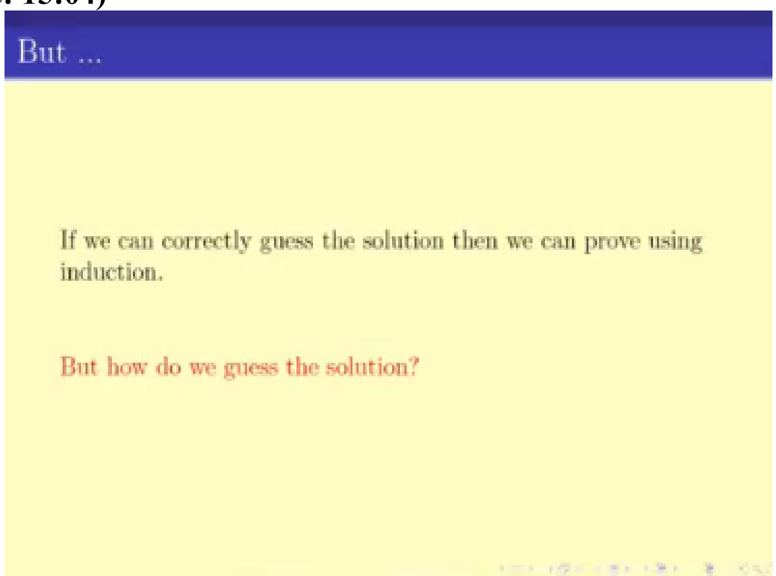


Again and again, we have to look at the base case, so base case n =1. So here,
H(1) = 2^1 – 1 = 1 which is right. So the base case is correct. So induction hypothesis say

$H(n) = 2^n - 1$, For inductive step, we have to prove, $H(n+1) = 2^{n+1} - 1$. Now,
$H(n+1) = 1 + 2H(n) = 1 + 2(2^n – 1)$ *(By induction hypothesis)* $= 1 + 2^{n+1} - 2 = 2^{n+1} – 1$, which is what we have to prove. So $H(n) = 2^n - 1$ for all $n \geq 1$. Note that, this is not only a way to proving the recurrence, this also if you go back to our previous video, this gives us a compact form for the number of moves required for the Tower of Hanoi problem.

So the Tower of Hanoi problem, therefore requires $2^n - 1$ moves and we got it by first modeling it as a recurrence relation and then solving the recurrence relation. Now how did you solve the recurrence relation, we first guess the recurrence relation and then we prove that the guess is right.

(Refer Slide Time: 15:04)



So this is how most of the counting problems work, you first model it a recurrence relation and then you solve the recurrence relation. But this is all this is fine, if you can guess the recurrence relations correctly. You first guess the recurrence relation and then prove it using induction. The main question is how do you guess the solution? And we will be doing this problem of how to guess the solution to the recurrence relation in the next video.

We will see one of the techniques and in the next few videos we will see the other techniques. Thank you.