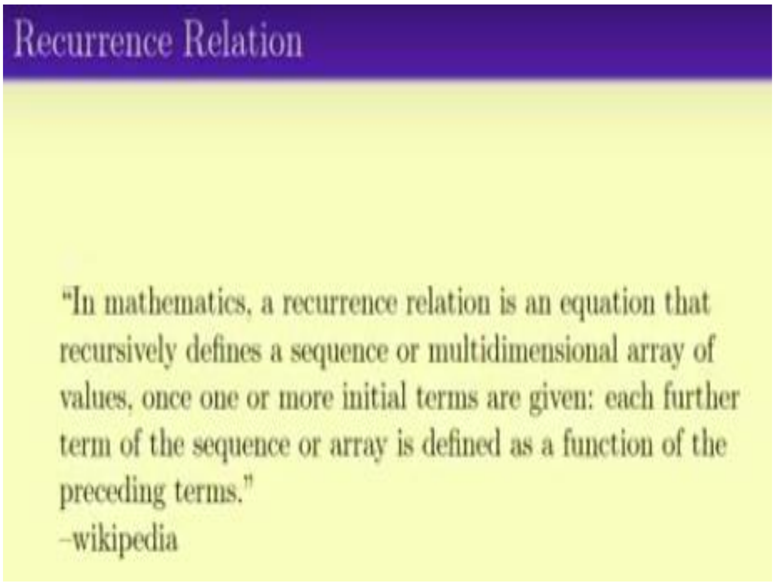
Discrete Mathematics
Prof. Sourav Chakraborty
Department of Mathematics
Indian Institute of Technology – Madras

Lecture - 40
Asymptotic Relations (Part 1)

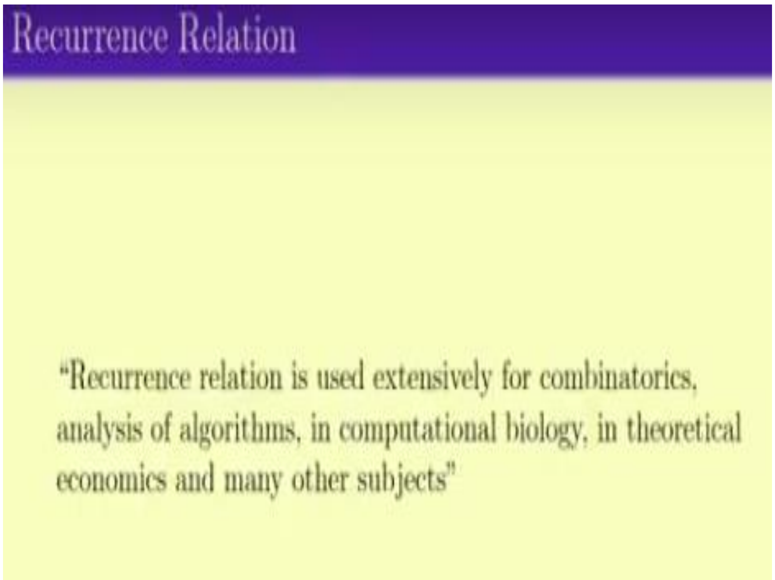
Welcome back, so we have been looking at recurrence relations.

(**Refer Slide Time: 00:04**)



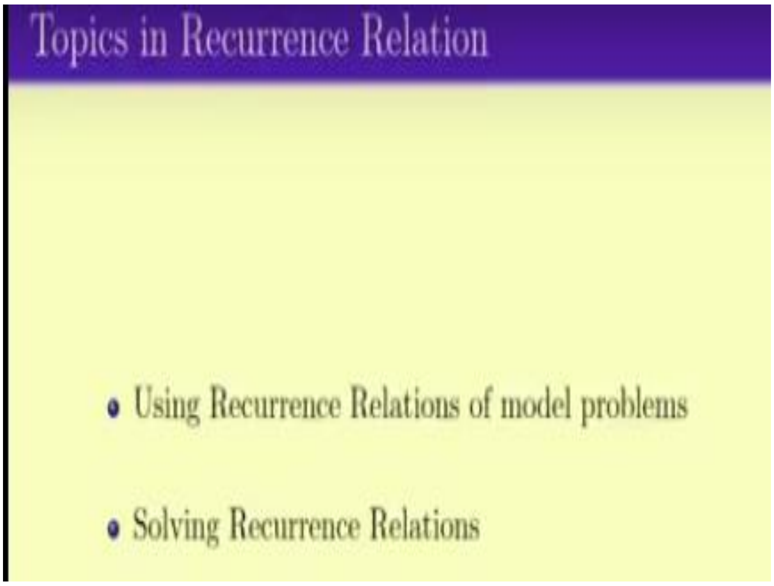
So what - the recurrence relation is a sequence of numbers where the initial set of numbers are given and the nth number or in this term in the sequence is given as a function of previous terms.

(**Refer Slide Time: 00:25)**



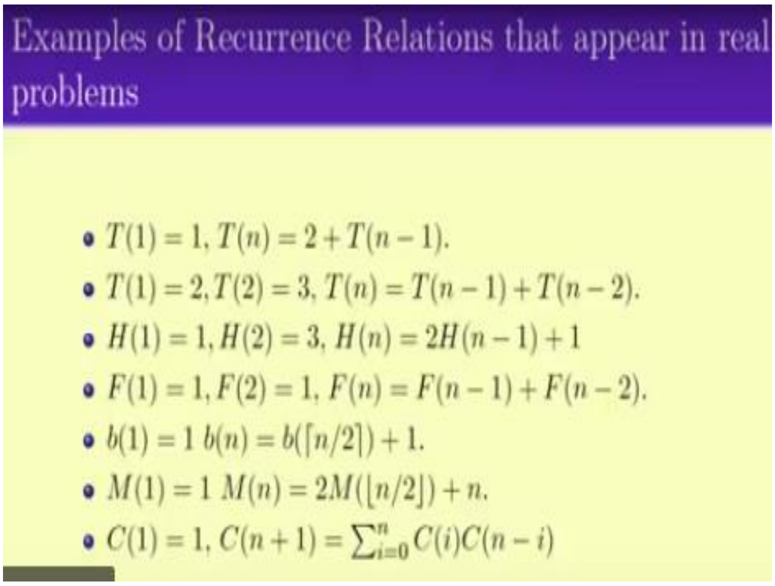
Now, recurrence relation is extensively used for combinatorics, analysis of algorithms and various other topics.

(Refer Slide Time: 00:34)



We have seen how recurrence relations can be used to model various counting problems and we were be looking at how to solve this recurrence relations.

(Refer Slide Time: 00:52)

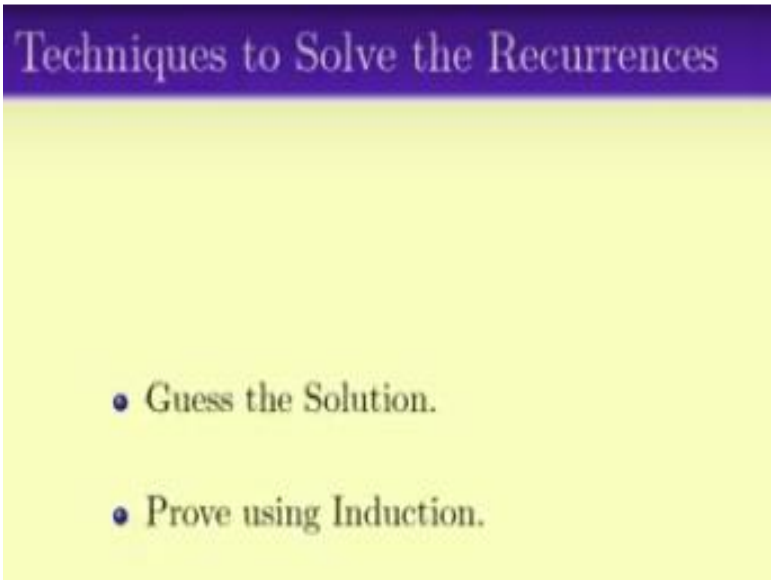


So, some of the examples of recurrence relations is that appeared are something like T1 equals to 1 and Tn equals to 2 plus Tn minus 1 or T1 equals to 1, T2 equals to 3 and Tn equals to Tn minus 1 plus Tn minus 2 or this one is what we get from the tower of Hanoi which is H1 equals to 1 H2 equals to 3 and Hn equal to 2 times Hn minus 1 plus 1.

Then, we have F sequence which is F1 equals to one, F2 equals to 1 and Fn equals to Fn minus 1 plus Fn minus 2, then we have something b1 equals to 1 and bn equals to b n over 2 plus 1, this is what we get from the binary search algorithm. And similarly we have this one what we get from the merge sort algorithm M1 equals 1.

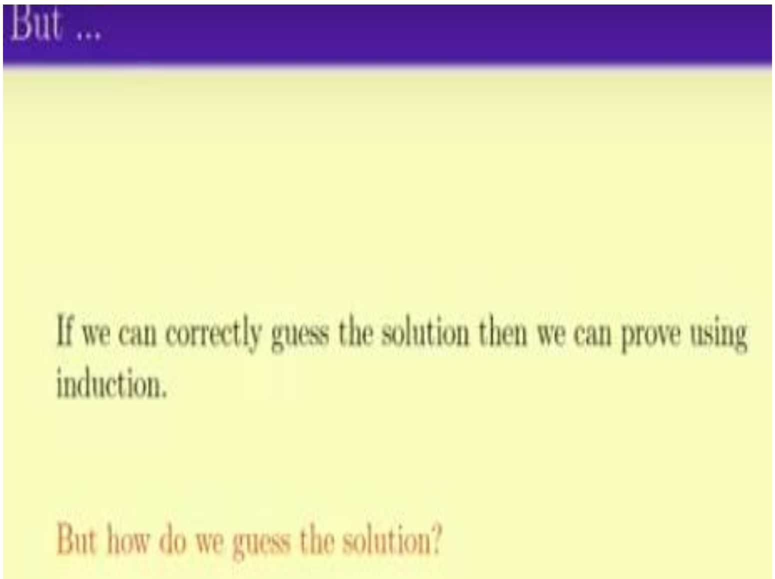
Mn equals to 2 times M n over 2 plus n and the one that we get Catalan number C 1 equals 1 and C n plus 1 equals to summation of i posed to 0 to Ci times C n minus i, Now the question that comes up is that how do we solve this recurrence relations? So we have seen somehow.

(**Refer Slide time: 02:19)**



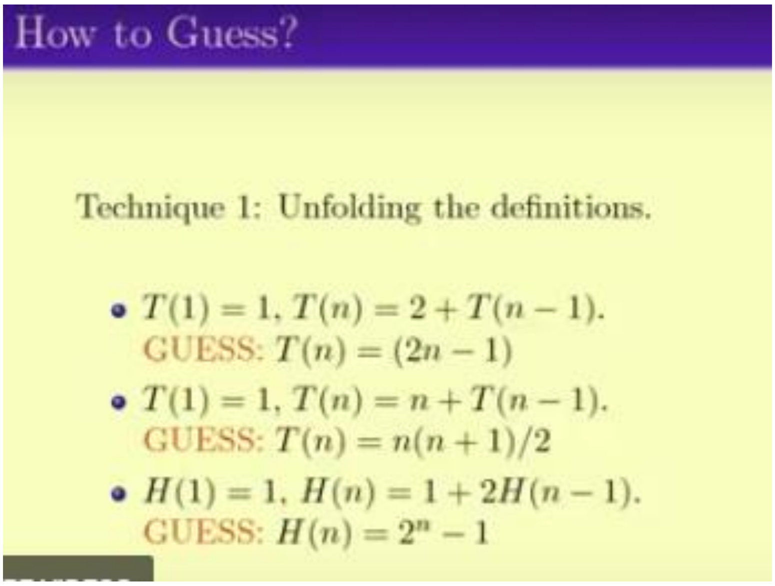
The technique that we have looked at is first of all, guess the solution and prove using induction.

(**Refer Slide Time: 02:22)**



So, in other words, if you correctly guess the solution then, we can prove using induction then the case is correct. The question is how do we guess the solution?

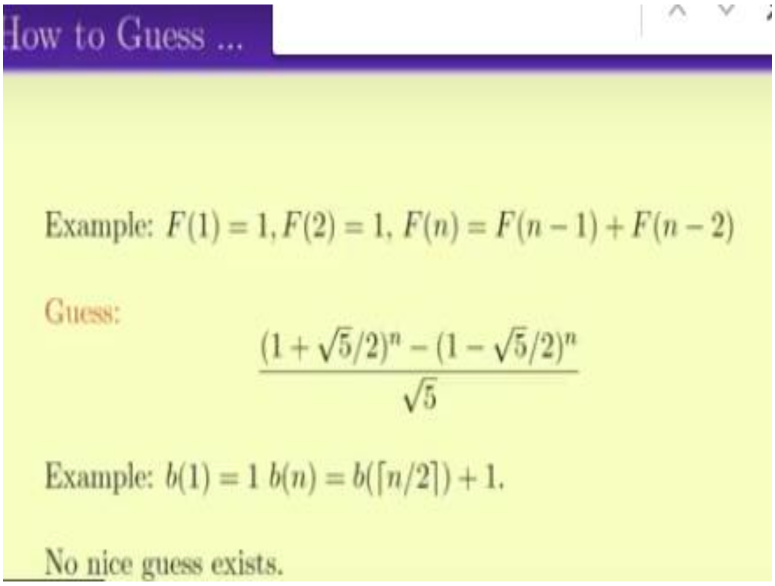
(**Refer Slide Time: 02:38)**



To the last two videos, we saw one particular technique namely we unfold the definitions and by unfolding definitions for example, we write T1 equals to 1 and Tn equals to 2 plus Tn minus 1, then we write T n minus 1 as T n minus 2 plus 2 and so on, and after ith iteration, I get Tn in terms of some expressions and Tn minus i and that helps us to find out choose the right i for (()) (03:11) we can get a full correct value of Tn and by doing so, we could guess that Tn equals to 2n minus 1.

In the next one, we could guess that it is equal to n into n plus 1 by 2 and in the Tower of Hanoi, we could guess that Hn equals to 2 power n minus 1. Now, any of these cases, the idea is guess and then prove by induction. Now problem is that many times guessing can become quite tricky.

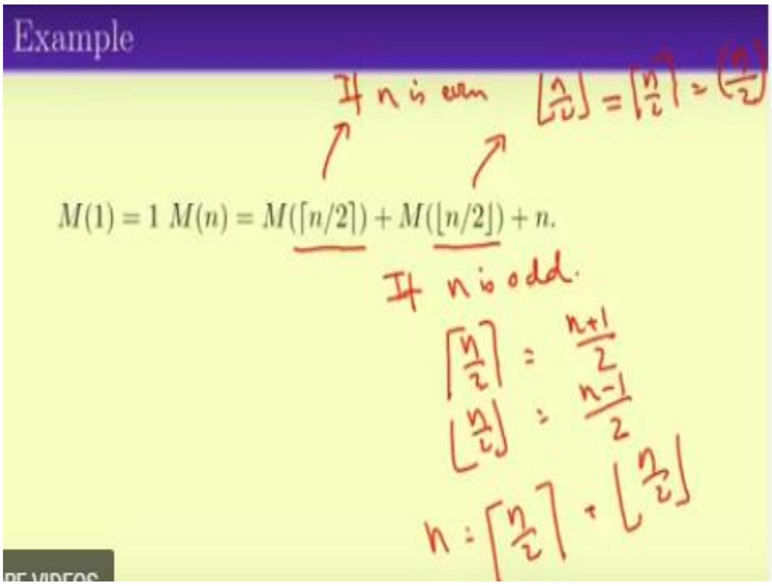
(**Refer Slide Time: 03:50)**



For example, if we have a Fibonacci expression like this, if F1 equals to 1 and F2 equals 1 and Fn equals to Fn minus 1 plus Fn minus 2. Now the guessing with value of Fn is quite complicated because the final value of Fn, the actual value is something as complicated as this, 1 plus square root 5 by 2 whole power n minus 1 minus square root 5 by 2 whole power n whole divided by square root 5.

You can clearly understand that guessing this value is not an easy job. Sometimes, we have expressions like this, b1 equals 1 and bn equals to b n over 2 plus 1 but because of this ceiling here, we can say that getting a simple clean expression for bn is not an easy job. In fact, there does not exist a neat case or nice case that exists.

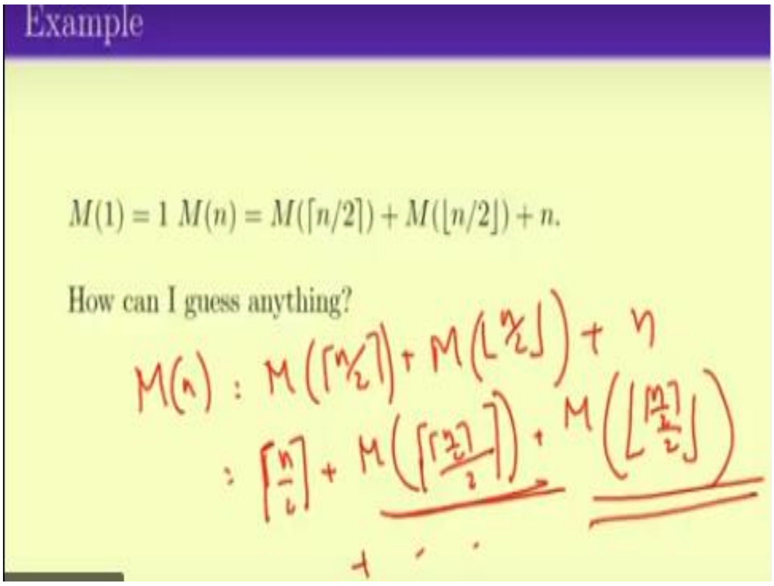
(**Refer Slide Time: 05:04)**



So how do we go about solving them. Let us look out an example, so here is an example that occurs in the merge sort algorithm, M1 equals to 1 and Mn equals to Mn over 2 plus Mn over 2 plus n where the first one is a ceiling, second one is a floor. What does the ceiling means? So if it is this means that this is the smallest integer that is bigger than n over 2 and this one is the largest integer that is less than n over 2.

So, if n is even then both of these n over 2 by 2 is equals to n over 2 which equals to just this n over 2 but if n is odd then n over 2, sorry, n over 2 is actually n plus 1 over 2 and n over 2 is n minus 1 over 2 and as you can see that n is equals to n over 2 plus n over 2 and this is the definition of these two expressions of floor and ceiling.

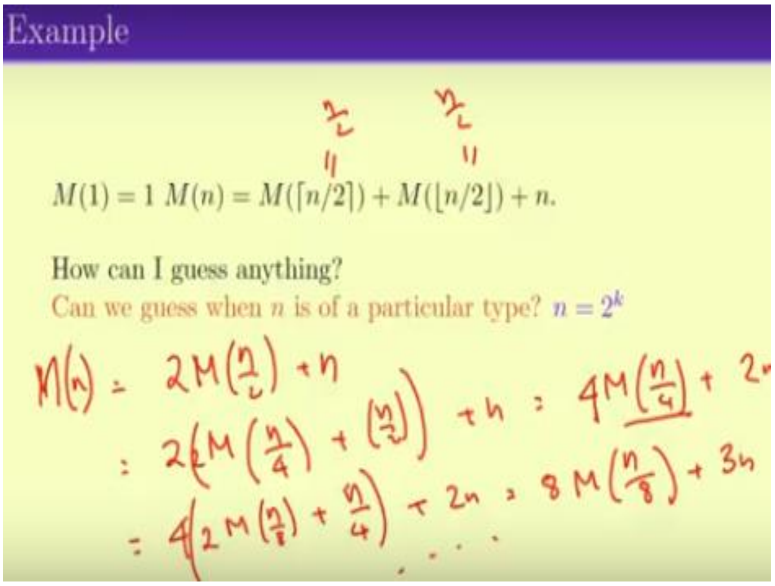
(**Refer Slide Time: 06:46)**



Now, how do we - Can you guess them. Can I guess anything? Of course, one of technique is to unfold it. Set unfold it. Mn equals to Mn over 2 plus Mn over 2 plus n which is again, I can write this one as n over 2 plus M floor of floor n over 2 by 2 plus M by 2 plus something, something. only (()) (07:36) this expression is becoming very (()) (07:38), not much that we can do about it.

The reason is that we cannot simplify this, even if any event okay, in that case, I do not have to write. This one I can, this is this plane n over 2 but then here this will become n over 2 by 2 ceiling of that and when is n over 2 by 2 ceiling. It is only way n over 2 is either or in other words, n is a multiple of 4. So, in other words, the only way I can simplify this whole expression is when n is a power of 2.

(**Refer Slide Time: 08:32)**



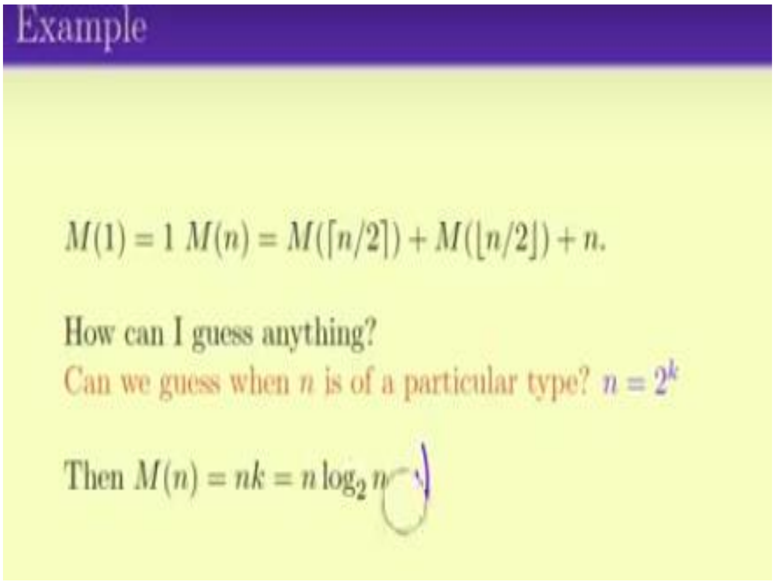
So if we can guess, question is that can we guess this number Mn? When n is a particular type and idea is that what is n equals to 2 power k. So let us see what happened. If n equals to 2 power k, the good thing is that both of them is just n over 2. Right? So I have Mn equals to so I get 2 i M n over 2 plus n which is equals to 2 times M. Now, n over 2 is also power of 2. Hence, also even number.

And hence I can just again apply this one and I get n over 4 plus n over 2 plus n which is 2 times M n over 4 plus 2n. What is the next one? Next one is 2 times this, so again 2 times, sorry I made a mistake here. It should be 2, so this should be 4. Right? So this should be 4. So what do I have again? 4 times when I expand this Mn over 4, I get 2i

M n over 8 plus n over 4 plus 2n which - Again you can see, here becomes 8 M n over 8 plus 3.

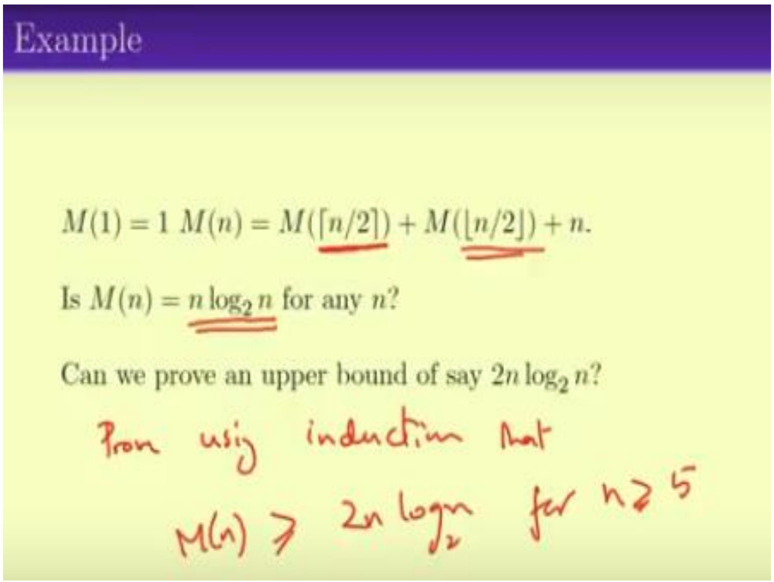
So, in other words, if I repeat it again and again and again after i (()) (10:47) I will get 2 power i M n over 2 power i plus i times n. This 3 is 2 power 3 and now the idea is that if you had put i equals to log n, so i equals to k. If i equals to k then, I have n equals to 2 power k. In that case, i equals to k, what do I get?

(**Refer Slide Time: 11:54)**



I get this term to become 2 power k plus k times M right? Which is, this is a, this is k plus 1 times n. So in other words, we have been able to do it, got guess it for a certain class of n which is namely for n equals to 2 power k. We could then get it to be nth of power k which is n power k. So if this should be plus 1 (()) (12:04) 2 plus 1. Good. So at least for some class of n, we can get.

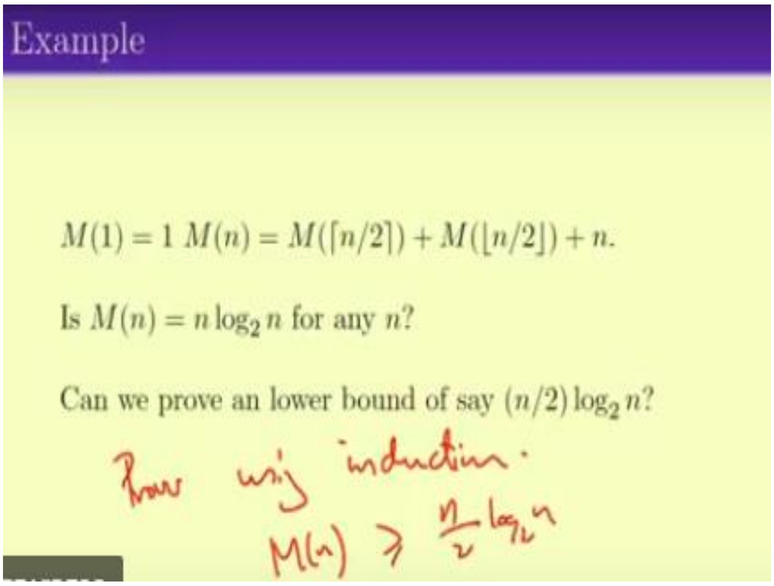
(**Refer Slide Time: 12:32)**



Question that comes now is that can we say that this is the right expression for all n and the fact is that possibly not. Because for non-normal n, there should be a floor and a ceiling which will just make this whole life more complicated. At least ugly took at with the calculus. Since that we ask - can get upper bound? Is it something like to n log n.

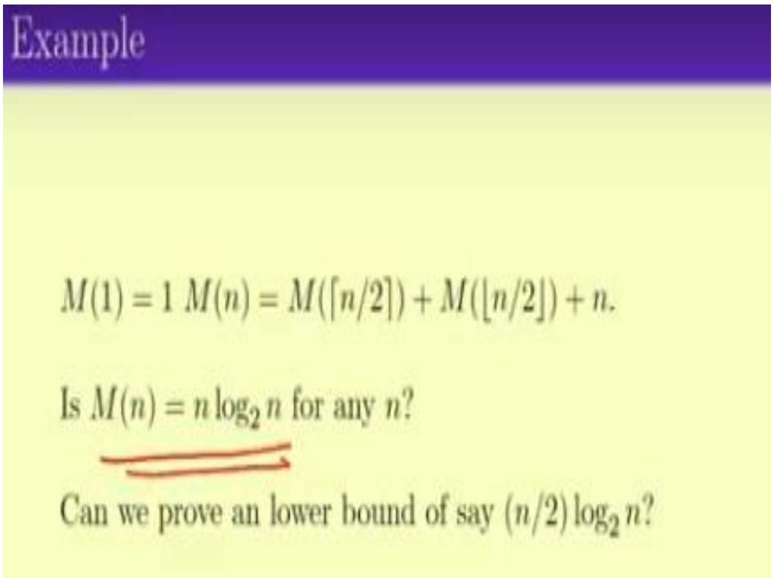
And I leave it to you, prove that prove using induction that Mn is indeed bigger than 2n log n for n bigger than 5. Okay?

(Refer Slide Time: 13:52)



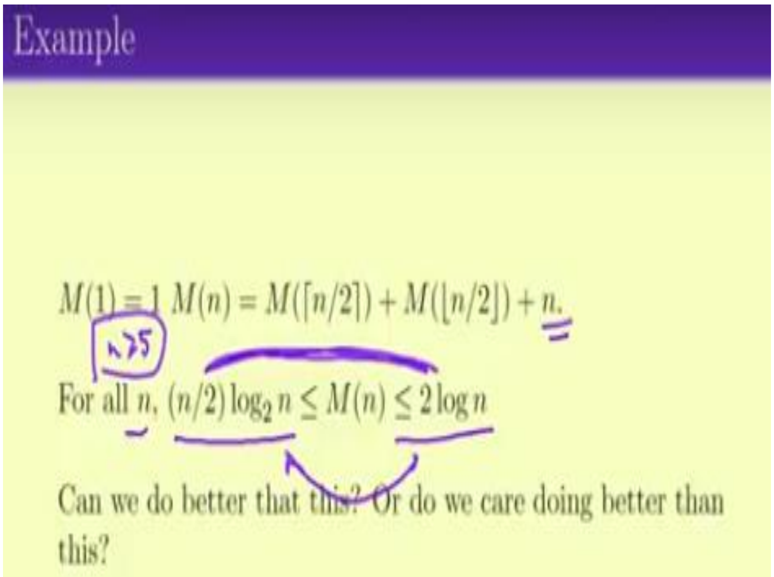
And so we can have proof that. But, this is just a upper bound. Can we come up with a matching lower bound and again it is so turns out that we can, we can prove that in lower bound it is n over 2 log n. Again, this is I leave it to an exercise, prove using induction, M of n is bigger than n over 2 log 2 plus n.

(Refer Slide Time: 14:26)



So what we have – so in fact, although we could not guess the actual value of Mn, we first just Mn for some class of n namely powers of 2 and then we could say that (()) (14:39) for all n greater than or equal to 5. Mn is lower bounded and upper bounded by two terms. It is upper bounded by 2 log n and lower bounded by n over 2 log n. So why we could not come up with the exact formula for Mn.

(Refer Slide Time: 15:17)



We could come up with the inequality upper bound and lower bound where the different between them is not too much. The upper bound is just 4 times this lower bound. So in many times, particularly for this example, so this the number of steps required for made by a particular program namely the merge sort. And when we have looking at the number of steps, we have not yet worried about two things.

First of all, what happens for small n. So we are only worried about big n. When the input size is big for that, so this is what is called as asymptotic. So, as n increases, what happens? For small n, we do not care and second thing is that we really do not care whether the bounds are right or not. In fact, as long as, we understand that the value that we are looking for is within a constant fraction of something else, we are happy.

So, in other words, of course, the question is that is always good to come up with an exact answer but if you cannot, we are usually happy, if I get an answer which is constant factor away. I will formalize all these things in the next class or next video, when we will be looking at what we know call asymptotic notations and using that we will formalize when we say that this is a pretty good estimate. Thank you.