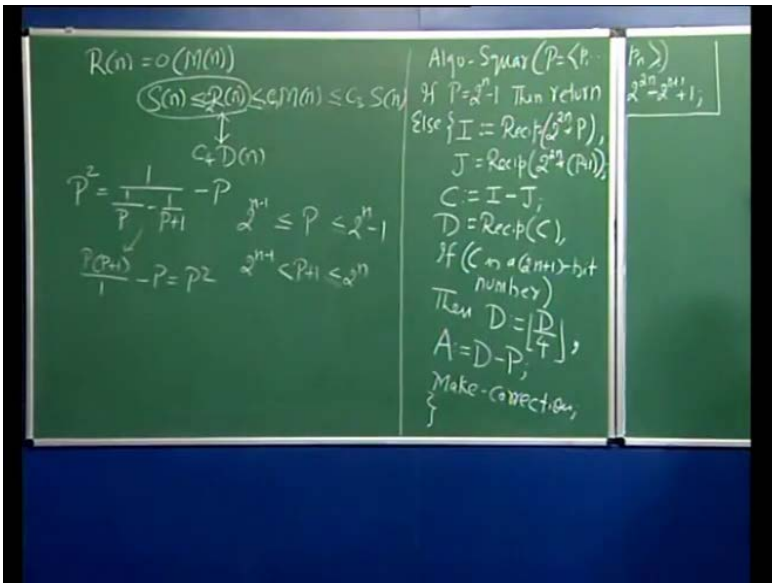
Computers Algorithms - 2
Prof. Dr. Shashank K. Mehta
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 18
Integer-Polynomial Ops II

Hello. So, in the last lecture we had shown the complexity of computing reciprocal of a number of n bits was of the order of multiplication of 2 n bit numbers.

(Refer Slide Time: 00:24)



This is what we had shown last time. Our long time goal is to show that with say some c 1 which this result shows, we also would like to show that this in turn for some c 2 is less than equal to square operation. So, squaring a number of n bit that is bounded by the reciprocal, we will also try to show for some c 3 that multiplication itself is bounded by square operations times. Therefore, all these three operations are within a constant from each other. It is not our objective to give an algorithm for these, but just to compare the complexity all basic operations of integers, that is to say multiplication reciprocal and square are within a constant time from each other.

Notice that these are equivalent with respect to some other coefficient of division that is to say that a reciprocal and division are also within a constant time from each other. That is trivial, because you can perform division by computing reciprocal and then multiplying by the numerator. The reciprocal is a special case of division. So, today what

I am going to show is, in this equality. I will try to show that we can bind the time of square by the reciprocal. Now, the basis of our today's discussion is the following observation, namely that P square is 1 over 1 over P minus 1 over P plus 1 minus P.

If you simplify this you are going to get P times P plus 1 over 1 and minus P, which is simply P square plus P minus P, which is simply P square. Hence, what we will try to do is that will try to construct the square of an n bit number by computing actually 3 reciprocals, that is 1 over P 1 over P plus 1 and 1 over the difference of it. This is the goal. Now, notice that when we deal with actual numbers of a certain size, we will have to adjust certain things. So things will be slightly different from what appears here, because, we do not have exact numbers such as 1 over p 1 over P plus 1. We can only approximate with certain accuracy. Now, I am going to write down the algorithm first. Then which is really motivated by this observation, we will try to show the correctness of it. Now, there is one small fact about P. We are assuming that it is an n bit number.

So, P itself is actually within 2 power n minus 1 and 2 power n minus 1, but if we have P equal to 2 power n minus 1. Then P plus 1 will become less than equal to 2 power n and strictly greater than 2 power n minus 1. The difficulty on this side is that, if it becomes equal to 2 th power n, then it becomes an n plus 1 bit number. This causes certain difficulty. So, this 1 lone case when P is 2 th power n minus 1. We can handle that separately and then we will assume that P is strictly less than this. This will make this strictly less than 2 power n, then both these numbers will remain n bit numbers. So, our algorithm, Algo-square takes P to be an n bit number, p 1 to p n as input.

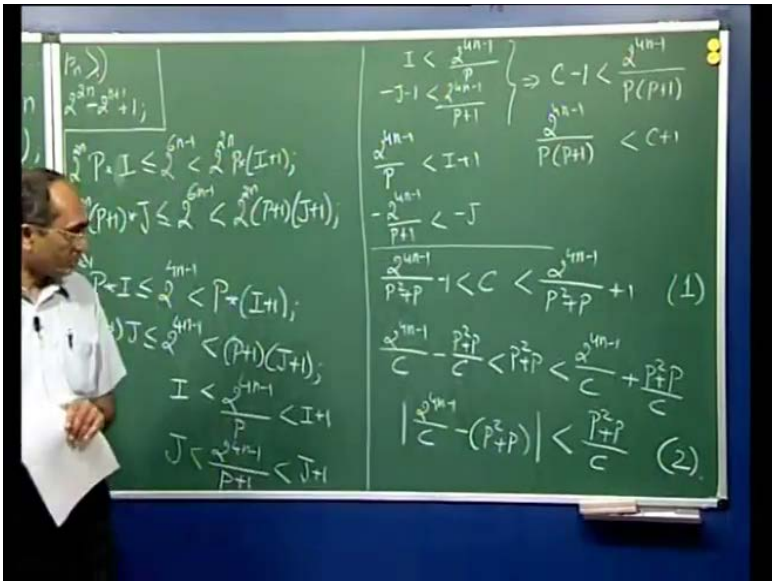
If P is equal to 2 power n minus 1, then return. We know that directly the square of this number this does not require much effort. We will return 2 power n plus 2 power n plus 1 plus 1. So, this 1 case we have separately handle. Else, we will first compute I which is reciprocal of 2 to the power 2 n times P. Notice, that instead of taking an n bit number, we make it as a 3 n bit number because we need a bit of accuracy in our numbers. Similarly, I will compute J which is reciprocal. This is the same algorithm that we have discussed in the last lecture, that is 2 power 2 n times 2 plus 1. Then, C is equal to I minus j.

Then, if we are computing D, D is reciprocal p of C. Now, here 1 special situation arises which we will understand later on, that is if C is a 2 n plus 1 bit number, then define D to

be D by 4. This will replace by D and will be replaced by D by 4. Now, effectively D is this number I am barring. Of course, whatever happens due to the boosting of the bits into the number, if we adjust those things D will be about this number. A will be our approximation for P square this minus P. That gives us it. Unfortunately, A may not be exactly P square. There is no room for approximation for P square, because it is an integer and P square exists within the representation range.

So, what we will have to do is, make some corrections. Later on, we will show what corrections are to made. But here we are going to leave some remark, make some corrections. Then return the value that should be the value of P square. So, now let us go about analyzing this algorithm. Now, as I mentioned we will assume our P and P plus 1 are n bit numbers. We will stay with that because this case has been separately taken out.

(Refer Slide Time: 09:45)



Now I, if you recollect from the last lectures is P times 2 power. Rather, 2 power 2 n times P times I is less than equal to 2 to the power. Notice, it is a 3 n bit number. So, we can do that as 6 n minus 1. This is what we expect the reciprocal should give me and should be strictly less than 2 power two n times P into I plus 1. This is the form… Then, what guaranty, we get from the reciprocal by the same taken 2 power 2 n P plus 1 times J is 2 power 6 n minus 1 which is less than 2 power 2 n P plus 1 J plus 1. These are from the fact that I, n and J are computed as the reciprocal of these numbers. I can simplify

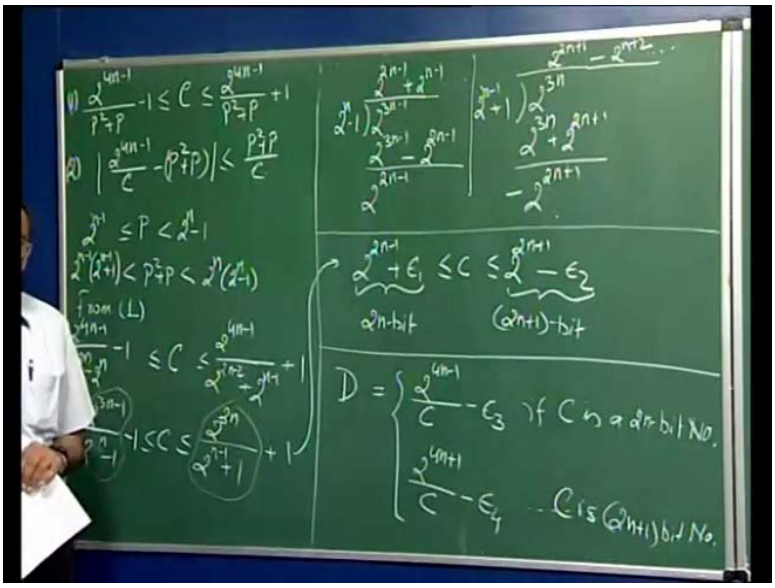
them. So, these are going to simplify to P times I less than equal to 4 n minus 1 less than P times I plus 1 or just cancel out 2 power n also from here.

So, you are going to get P plus 1 times J less than equal to 2 to the power 4 n minus 1 less than P plus 1 J plus 1. Now, let us try to find a bound for I from this. So, I am going to write down a bound for I, as I is less than 2 to the power 4 minus 1 over P. Here, we have I plus 1 J is 2 to the power 4 n minus n by P plus 1 less than J plus 1. Now, our C which is I minus J, we can now bind from the above and below in terms of these terms. So, let see what does that give us? It gives us minus J minus 1 is less than minus of this. So, I is less than 2 to the power 4 n minus 1 by P and minus J minus 1 is less than A to the power 4 n minus 1 by P plus 1.

Combine these two and you get I minus j which is C minus 1 less than this, is minus this, is minus this and minus this which is 2 to the power 4 n minus 1 into 1 over P minus 1 over P plus 1 which is nothing but, P into P plus 1. Now, if I use the other inequality, I will get 2 to the power 4 n minus 1 by P is less than I plus 1 and minus 2 to the power 4 n minus 1 over P plus 1 is less than minus J. Again, if I add the 2, I am getting C plus 1 to be less than 2 to the power 4 n minus 1 over P into P plus 1. We get a bound above and below for C. Combining the two, we have our first inequality C is less than 2 to the power 4 n minus 1 by P square plus P plus 1 and the other in equality is less than 2 to the power 4 n minus 1 over P square plus P minus 1.

We have these 2 bounds. Let us say, I am going to call this equation 1. Now, if I multiply the whole thing by P square plus P over C and simplify this, this thing turns out to be, let us just write down here, 2 to the power 4 n minus 1 over C minus P square plus P by C less than P square plus P. Notice, I am multiplying by P square plus P divided by C in the entire thing. So, this becomes P square plus P over here. This becomes 2 to the power 4 n minus 1 over C plus P square plus P by C. So, we get 2 to the power 4 n minus 1 by C minus P square plus P mod is less than P square plus P over C. This is my equation number 2. Okay, so let me just rewrite the in equality that we had which is 2 to the power 4 n minus 1 over P square plus P.

(Refer Slide Time: 17:06)



I think we made a mistake. We should have had less than or equal to here. C is not strictly less than 2 to the power 4 n minus 1 over P square plus 1. This was our first equation and the second one we had was mod of 2 to the power 4 n minus 1 over C minus P square plus P is less than or equal to P square plus P over C. Now, let we know that the range of P itself is in the range 2 to the power n minus 1. So, P square plus P, that is P into P plus 1. That will become 2 to the power n times 2 to the power n minus 1 here.

Over here, we will have 2 to the power n minus 1 into 2 to the power n minus 1 plus, so that it gives me an open lower and upper bound for P square plus P. I am going to plug this in and get another bound for C. So, from equation 1, what I find is that C on this side is 2 to the power 4 n minus 1. I am going to plug in the larger of the 2 values that is 2 to the power 2 n minus 2 to the power n minus 1. Over here, 2 to the power 4 n minus 1 by 2 to the power 2 n minus 2 plus 2 to the power minus 1 and plus 1 can be simplified straightly. So, from 2 to the power 3 n minus 1, 2 to the power 2 n minus 2 and C to power 2, I am going to remove n minus 1.

So, it is 3 n over 2 to the power n minus 1 plus 1 plus 1. Yes, 2 to the power this one, we got it. This is just 2 to the power n. My mistake is that this is just 2 to the power n minus 1. Yes. So, this is this minus 1. You are right. I would like to now estimate this number and similarly this one. So, let us see what happens? This is 2 to the power 3 n minus 1

divided by 2 to the power n minus 1. I can take 2 to the power 2 n here. I will get 2 n minus 1 rather. Well, I got to clean it up. Let me retry. This is 2 to the power n minus 1. Dividing 2 to the power 3 n minus 1, this is going to be 2 to the power 2 n minus 1.

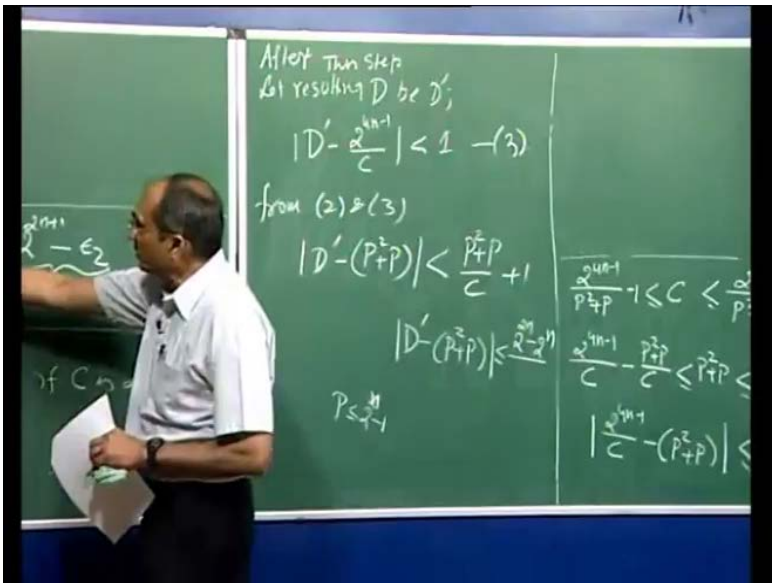
This gives me 2 to the power 3 n minus 1 minus 2 to the power 2 n minus 1. I have 2 to the power 2 n minus 1 left. So, that plus 2 to the power n minus 1 and something. We will just stop there and on the right hand side, we have 2 to the power, there n to be divided by 2 to the power n minus 1 plus 1. That means, we have to take in the quotient, 2 n plus 1. So, we have 2 to the power 3 n plus 2 to the power 2 n plus 1. We get minus 2 to the power 2 n plus 1. So, I am going to have a minus 2 to the power n plus 2 n plus 2. We will stop here again.

So, there is something more here, the lower terms. So, this inequality becomes 2 to the power 2 n minus 1 plus some epsilon. Notice that, this one is not much comparable with what we have here. So, we can very much ignore the importance of this. We have a larger positive term there. So, we have some epsilon 1 c 2 to the power 2 n plus 1 minus epsilon. We have 2 negative terms here. Once again, because of that large negative term a plus 1 does not really matter. What you notice here is, this number is 1 followed by 2 n minus 1, zeros and some small number. This will be a 2 n bit number. This is 1 followed by 2 n minus 1 and zeros.

So, this n plus something is not going to reduce the size of the number. On the other hand this alone is 1 followed by 2 n plus 1 0, but the moment you subtract something, 1 bit goes down. So, this becomes only 2 n plus 1 bit number. What we have here is 2 possibilities for C. Either C is 2 n bit number or 2 n plus 1 bit number. These are the 2 possibilities we have to deal with separately, okay? Now, in our program we are computing D, which is the reciprocal of C here. But, depending on the size the reciprocal, we will compute 2 different numbers. So, D will be 2 to the power 4 n minus 1 by C minus C 3, if it is 2 n bit number.

It is going to give m 2 to the power 4 n minus 1 over C minus some small fraction here, say we have epsilon 3 here, if C is a 2 n bit number. Over here, because it is a 2 n plus 1 bit number, we will have 2 to the power 4 n plus 1 by C minus epsilon 4, if it is 2 n plus 1 bit, C is bit number. These are the two possibilities. Now, therefore, whatever algorithm does is that, in this case we divide D by 4.

(Refer Slide Time: 26:51)



So, If you recollect the algorithm had a step here which said, if C is a 2 n plus 1 bit number, then we computed D to be D divided by 4. We have done this. So, let this was a step number. So, after this step, let resulting D be D prime. Let us suppose, D prime denotes the number that we compute after this step. So, D prime will remain this. If C is a 2 n bit number, then D prime will become this divided by 4. If it is a 2 n plus 1 bit number, so what we notice here is that, D prime minus 2 to the power 4 n minus 1 over C is less than 1. We notice that, this is less than o1. Our error is within 1.

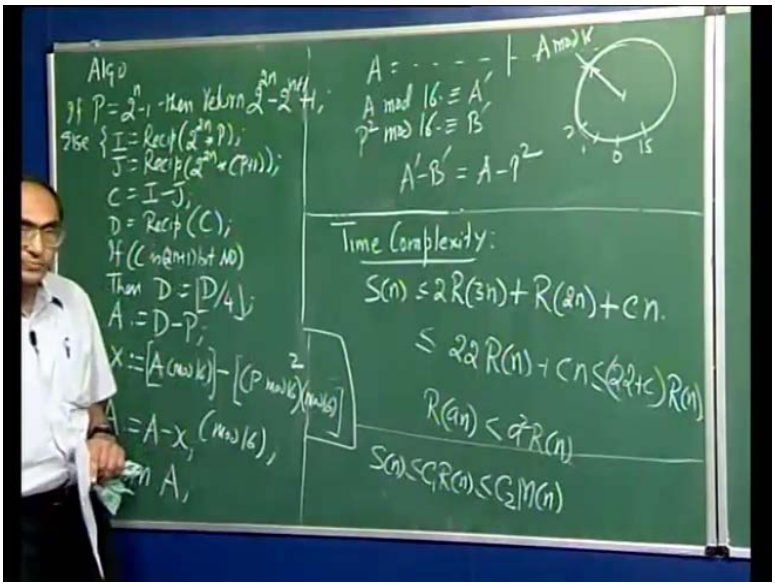
So, I will label this equation as equation number three. So, we have 3 crucial equations that we want to use, 1, 2 and that one 3. So, from 2 and 3, let us take a look at these 2 equations carefully. One of the term is same in both. Now, imagine you have a real line over here. We have the term 2 to the power 4 n minus 1 divided by C. Then, D prime is within this. This is plus 1 and minus 1. From this reference, this is where my D prime. It falls somewhere here. D prime falls within plus minus P square plus P by C. So, over here you have plus P square plus P by C and minus P square plus P by C. In this range, occurs P square plus P.

So, because this is the difference between these 2 bounded by P square plus P by C within this range, hence the difference between D prime and P square plus P D prime is in here. P square plus P is somewhere. So, the difference between D prime and P square plus P must be at most P square plus P by C plus 1, in absolute sense. So, let us try to

state that we have from 2 and 2, the difference between D prime and P square plus P. It is less than P square plus P by C plus 1. That is the most that is possible between these 2. So, now we have this in equality for C. I need a lower bound .We had a lower bound here. It is 2 by 2 n minus 1 plus something.

So, we can say that this difference D prime minus P square plus P is bounded above by P. Of course, this is 2 to the power n minus 1. So, this becomes2 to the power 2 n minus 2 to the power n. That is the upper bound for P square plus P divided by a lower bound for C is 2 to the power 2 n minus 1, 2 power 2 n minus 1. Then, overall we have a plus 1. This quantity is of course, less than 3, because this leaves only 2 minus something plus 1. So, we have the overall difference between what we have computed and what we want to compute is at most 3. Now, let us just go back and recollect our algorithm which I will rewrite.

(Refer Slide Time: 32:22)



Again, Algo says if P is 2 to the power n minus 1, then return 2 to the power 2 n minus 2 to the power n plus 1 plus 1, else we had I reciprocal of 2 to the power 2 n times P. J was a reciprocal of 2 to the power 2 n into P plus 1. C is I minus J. D is the reciprocal of C. If C is 2 n plus 1 bit number, then replace D by D divided by 4. Then, we say A is given by D minus P. At this point what we have, notice that D is what we had pointed out as C as D prime here. A is D prime minus P. So, this is pointing out that the difference between

A and P square is at most 3. So, at this point we would like to correct A, because the error might be up to 3.

How do we do that? The problem is, it is P square that we want to compare A with. We cannot compute P square, because all we are allowed to do is to compute the reciprocal. So, there is a nice trick we can apply. What we will do is that, if you look at A in terms of bits, what we have here is that if I cut off A up to some point. In other words, if I compute A mod 16 and I compute, say P square mod 16, then if I compute these 2 numbers, if they are equal then their mod will be equal, of course. But, in case they are off, then they are at most off by 4, may be 3. So, what will happen is that, here the difference between these two numbers is, let us say A prime and this is say B prime.

Then, the difference A prime minus B prime is also the difference between A and P square, because we know that the difference does not exceed 3. See, the only information I will lose, if the difference was the multiple of 16. But, in A, if this is my 0 and if I compute 1, 2 and so on and 15, in this cycle our error can be positive or negative. Of course, so if somewhere here this is where A, this is where A mod 16 is pointing in this clock. Then, up to 3 points here or up to 3 points here, it will be the place for P square mod 16, because it is plus or minus. We do not know, but that will be visible in mod as well. We will not lose, because this cycle is of 16 size.

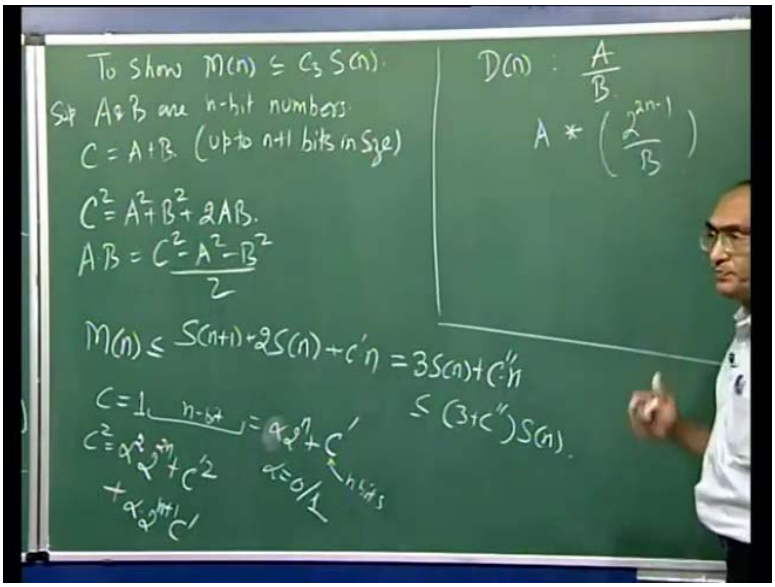
Hence, it is good enough to compare A mod 16 and P square mod 16. Whatever error comes, that become add or subtract to A to get exactly the value of P square. So, what we are going to do is to compute X to be A mod 16 minus P modulo16 square mod 16 and the whole thing mod 16. Then, we are going to give A to be A minus X. This is the exact error between A and P square as we have argued. Hence, we have A minus P square. We have adjusted, so at this stage we can now return A. This must be exact value of P square now, times to estimate the time complexity of this algorithm. In this, we have computed reciprocal of two 3 n bit numbers.

These are both 3 n bit numbers. So, I will just use S n. S n is the time to square. It is less than 2 times reciprocal of 3 n plus reciprocal of 2n plus Cn. It is 2 n bit or 2 n plus 1 bit. So, it is R 2 n plus some linear terms, because, we are subtracting. So, here as well as here, this is what is an upper bound for our whole process. The reciprocal, we have

already shown is as hard as multiplication which is at most square quadratic. So, this can be less than 18 of R n and this is at most 4 of R n.

So, the whole thing is 22 of R n plus some C. We n notice that, I have used the fact that R of a n is less than a square of R n. This is of course, linear. So, this whole thing can be written as 22 plus C R n. So, this was one of the objectives in this series of an equalities that we wanted to establish that S n is bounded linearly by R n. So far, what we have accomplished is S of n is less than equal to some C 1 of R n which is less than equal to some C 2 M of n. These 3 things are done. Now, a very simple argument can show that this is bounded by S n.

(Refer Slide Time: 41:50)



So, let us try to show that M of n is less than equal to some C 3 of S n. Suppose, you have say a suppose A and B are n bit numbers. So, let us take some C equal to plus B which can be up to M plus 1 bit in size up to n plus 1 bits n size. Now, if I square this, so C square is A square plus B square plus 2 A B. So, this product A B is nothing but, C square minus A square minus B square over 2. So, I can compute the product of 2 n bits numbers by squaring an n plus 1 bit number and 2 n bit numbers. So, what we have shown here is that, M n is less than equal to S of n plus 1 plus 2 of S of N plus some C prime time n.

Now, this C which is say n, if it is a n bit number, then there is no problem. This will also be S n, if it is an n plus 1 bit number. Then, it will look like 1 plus an n bit number. To

square this, this will be 2 to the power n. So, I will write down this as some alpha 2 to the power n plus C prime which is an n bit number and alpha is 0 or 1. So, C square is nothing but, alpha square which is again 0 or 1, 2 to the power 2 n plus C prime square plus alpha times 2 to power n plus 1 C prime. This involves only 1 squaring of n bit number and others are linear operations. Hence, this can be reduced to some 3 s n plus some C double prime n, which can again be in notice that S n is more than linear. So, this I can write down as 3 plus C double prime S of n. Hence, we again have some C 3 and S n.

So, what this whole thing says that the difficulty of all these 3 operations are equal. They are within a constant factor from each other and the last thing that I had mentioned at the start also, is that if D n is division of an n bit number by another n bit number. This can be estimated by first computing A and then multiplying it to a reciprocal which is 2 to the power some 2 n minus 1 by B. Multiply these 2 and then adjust your bits decimal points to get A by B. So, this involves 1 reciprocation and 1 multiplication. Hence, the difficulty of division is also of the same order as this.

Hence, all the base basic integer operations on an n bit number are of equal difficulty. This is the objective of our discussion and this is what we have established. In the next lecture, we are going to show that similar kind of relation exist on polynomials on single variable, because the ides are very similar and we will be able to show the similar kind of relation.