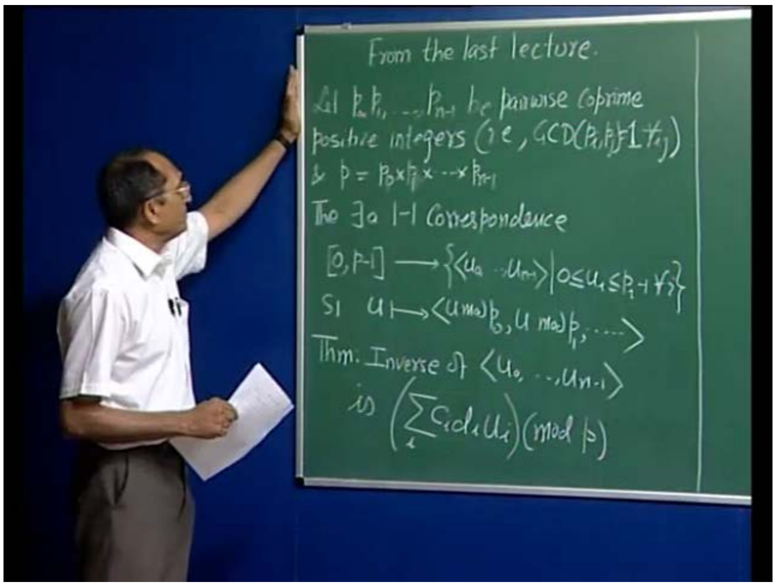
Computer Algorithms – 2
Prof. Dr. Shashank K. Mehta
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 21
Chinese Remainder II

In the last lecture, we had introduced Chinese remainder theorem.

(Refer Slide Time: 00:20)



From the last lecture.

Let $p_0, p_1, \ldots, p_{n-1}$ be pairwise coprime positive integers (i.e, $GCD(p_i, p_j) = 1 \; \forall_{i,j}$)

& $p = p_0 \times p_1 \times \cdots \times p_{n-1}$

The $\exists$ a 1-1 Correspondence

$[0, p-1] \longrightarrow \{\langle u_0 \ldots u_{n-1} \rangle \mid 0 \leq u_i \leq p_i - 1 \; \forall_i\}$

St $u \longmapsto \langle u \bmod p_0, u \bmod p_1, \ldots \rangle$

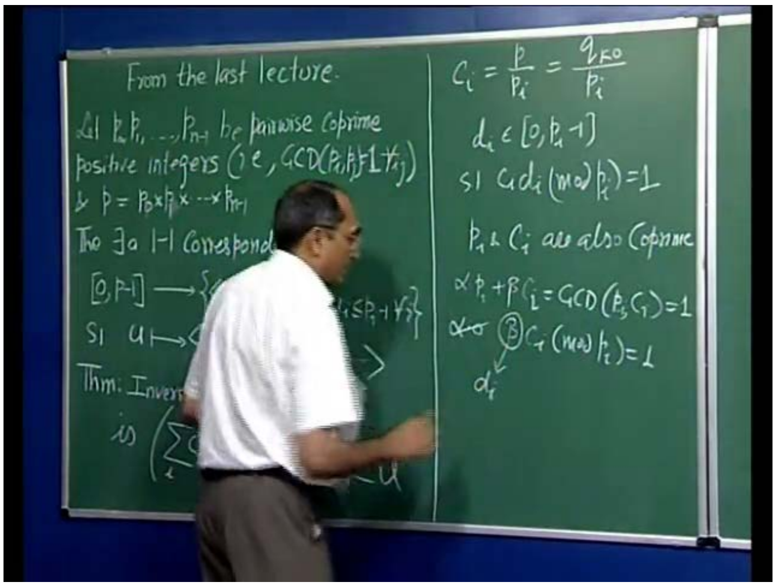
Thm: Inverse of $\langle u_0, \ldots, u_{n-1} \rangle$

is $\left(\sum_i c_i d_i u_i\right) (\bmod p)$

Let me review quickly what we have is n positive integers, which are pair wise co prime. It means that the greatest common divisor is one for every pair, let p be the product of this integer. Then, the remainder theorem says that there is one to one correspondence between all the integers between from 0 to p minus 1. Such end touples where each integer is between 0 and or highest integer is written 0 and p i minus 1. The correspondence is that for each u, there is a unique touple and touple where the highest element of this touple is v reminder u on division by p i.

We also showed that the converse so, computing. We have shown how to compute, efficiently compute with end touple from the given integer u and then, towards computing the converse. Suppose, we have given an end touple belonging to this set. Then, we show that this quantity is equal to u where the remainder of u with respective p i is this u i.

(Refer Slide Time: 02:00)


From the last lecture.

Let $p_0, p_1, \ldots, p_{n-1}$ be pairwise coprime positive integers (i.e., $GCD(p_i, p_j) = 1 \; \forall i, j$) & $p = p_0 \times p_1 \times \cdots \times p_{n-1}$

The $\exists$ a 1-1 Correspond

$[0, p-1] \longrightarrow \{$ $\leq p_i - 1 \; \forall i\}$

s.t. $u \longmapsto <$ $>$

Thm: Inver

is $\left(\sum_i \right.$ u

$C_i = \frac{p}{p_i} = \frac{q_{k0}}{p_i}$

$d_i \in [0, p_i - 1]$

s.t. $c_i d_i \pmod{p_i} = 1$

p_i & c_i are also coprime

$\alpha p_i + \beta c_i = GCD(p_i, c_i) = 1$

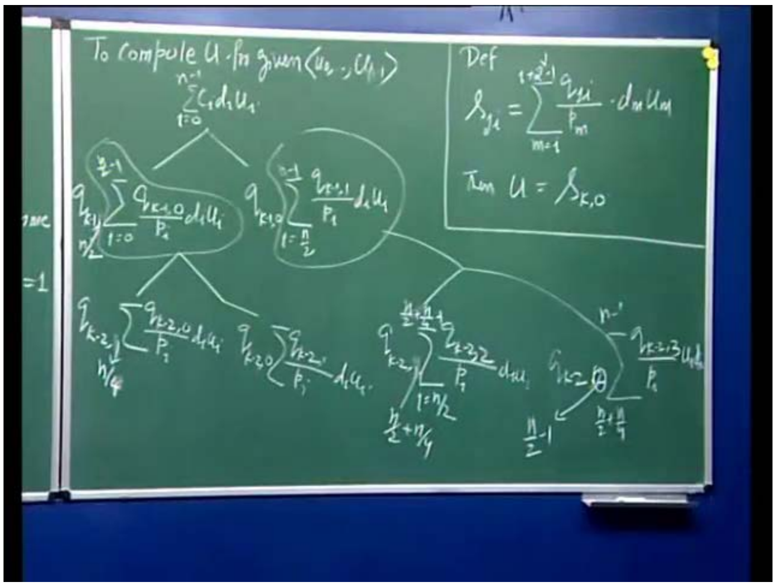
~~α~~ $\textcircled{\beta} c_i \pmod{p_i} = 1$

d_i

Over here c i was nothing but p by p i, which is same as q k 0 divided by p i, which is the product of all the integers. Other p i and d i was supposed to be in the range 0 to p i minus 1 such that c i d i mod p i is 1. In other words, under the module of computation d i is the inverse of c i. Given this, we had shown this what the u is. We also arguing that always with the d i exist simple because all these are mutually coprime. We showed quickly, let me recap that p i and c i are also coprime. From that, since every GCD can be expressed as some alpha time p i plus beta c i equal to GCD of p i c i that is to say that it exists, some alpha. Beta says that this combination is GCD which is 1.

Now, if you compute modular p of both sides, we get alpha times modulo p which is 0. Hence, we can ignore this. We are getting beta time c i module of p i is 1. So, this was our d i. Now, we are going to assume that d i's are pre computed. Remember, if you are going to perform such computations, re computations repeatedly you have pre computed d i. Now, let us take a look at computations of u, given u i.

(Refer Slide Time: 04:39)



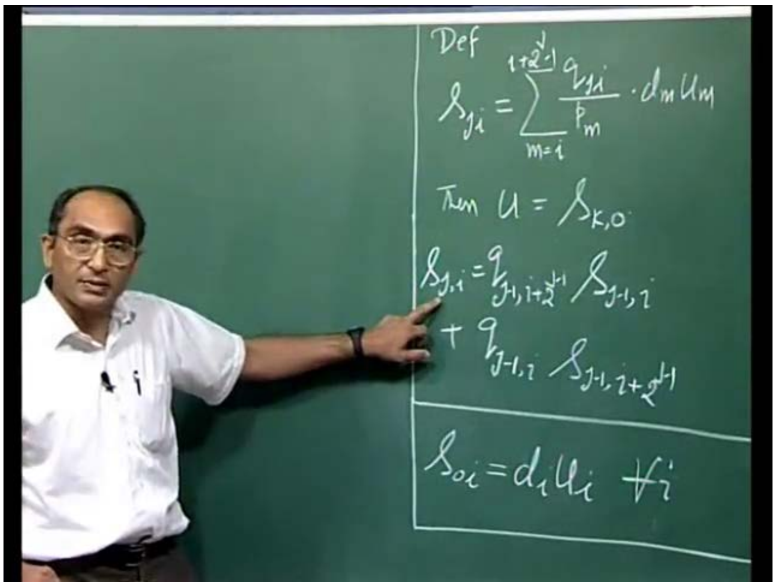
So, the problem to compute u of given touple u 0, u, u n minus 1. Now, if you look at this sum, you notice that this could be broken as c i d i u i i going from 0 to n minus 1. I can split this is all the p i multiplied except b j is multiplied except c i. So, I am going to split this sum into middle, at middle. We have, i going from 0 to n by 2 minus 1. All this cases, this c i will have p n by 2 in this factor p n by 2 plus 1 will be present in this factor. All those will be normal.

So, I am going to take that out. I can express that is q k minus 1, 1. This part will be now q k minus 1, 0 divided by p i d i u i. This side will look like q k minus 1, 0. Some will go from i equal to n by 2 minus 1 q k minus 1 divide by p i d i u i. Notice that, if you multiply this into this, you will get precisely u k 0. This is being divided by p i. But, all these p i in this part do not occur in this. They all occur in this. So, I can actually separate this out. Similarly, these p i occur in q k minus 1, 1. So, I can separate this quota.

Similarly, I can go breaking these sums. This time, if want to break this sum, I can split this as sum q k minus 2, 1 sum q k minus 2, 0 over p i d i u i. This side will be q k minus 2, 1 by p i d i u i. This side that splits to as q k minus 2, 3 sum q k minus 3, 2 over p i d i u i. The other sum looks like q k minus 2, 2. This will my, i here will go from n by 2, 2 n by 2 plus n by 4. Here, it will go minus 1. This will go from n by 2 plus n by 4 n minus 1 q k minus 2, 3 over p i u i p i. Now, I can keep on reducing this.

So, what I notice is I could define. Let me define sum as S j i as q j i over p m d m u m. m goes from i plus 2 power j minus 1. If we define this then, our desired u is nothing but S k, 0. This is what we want to compute. Now, I need to write a recurrence relation. So, we have always been using the second index as the index in the adding. So, I should put this as a n by 2, which is n by 2. This is fine. This will be n by 2 minus 0. This should be n by 2 plus n by 4. Over here, we will write n by 2 minus 1. This is only up to n by this n by 4. This is fine. So, that is right. Now, let us try to write the recurrence relations along. So, what we can do.

(Refer Slide Time: 11:53)



Def

$$S_{ji} = \sum_{m=i}^{i+2^j-1} \frac{q_{ji}}{p_m} \cdot d_m u_m$$

Then $u = S_{k,0}$

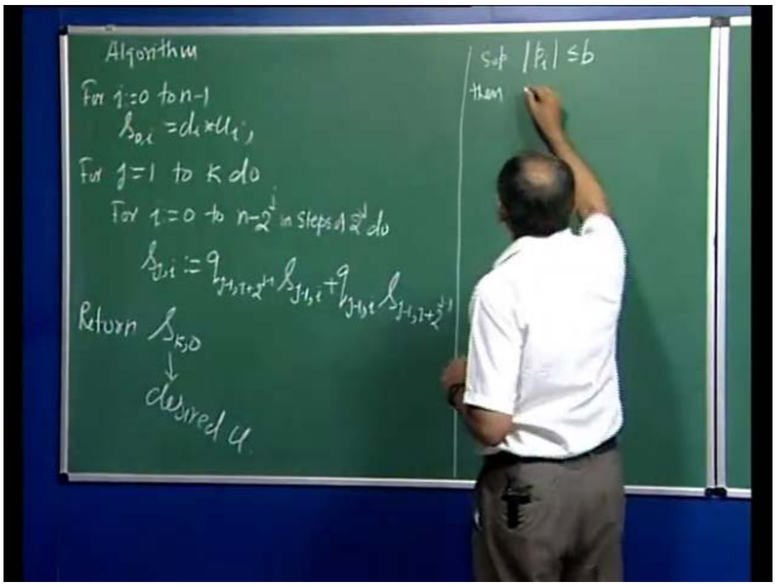
$$S_{j,i} = q_{j-1,i+2^{j-1}} S_{j-1,i} + q_{j-1,i} S_{j-1,i+2^{j-1}}$$

$$S_{0i} = d_i u_i \quad \forall i$$

Notice that this s j i j i can be broken into parts of half side just the way we have broken here. We will get q j i plus 2 power j minus 1. Sorry, this will be j minus 1. So, we need some more space s j i is q j minus 1 i plus 2 power j minus 1 S j minus 1 i plus, here we will have q j minus 1 i S j minus 1 j plus 2 power j minus 1. We have this recurrence, which is what exactly I have shown here. Now, this gives us the way to start from the bottom and build our way up.

So, I need to also give you the base cases. That is 0 i, this is nothing but there is only one term in this sum. That product is just p m and m divides. So, this is gone. All you get is d i u I, for all i. Now, we have the base cases. We have the recurrence. Our goal is to compute this. So, our algorithm will begin with assigning these values as this. Then, each will give us the higher level S. Finally, will output this half level may be S k naught.

(Refer Slide Time: 14:19)

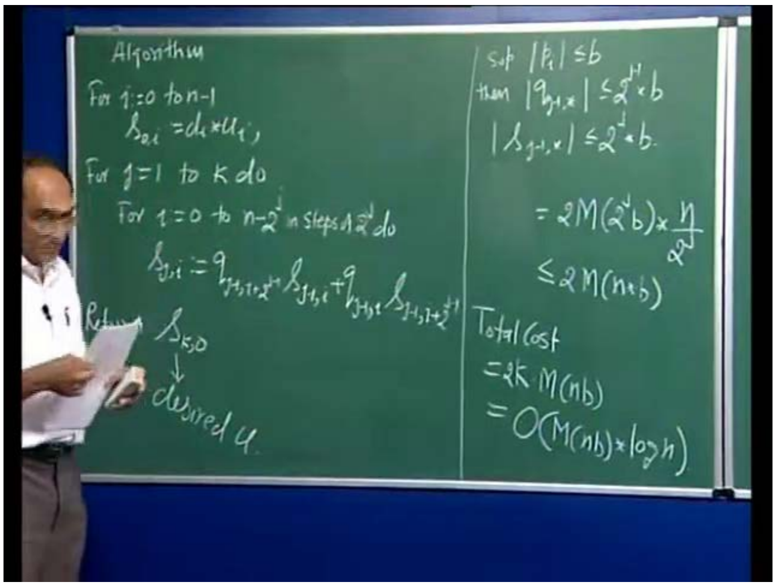


```
Algorithm
For i := 0 to n-1
    $s_{0,i} := d_i * u_i$;
For j = 1 to k do
    For i = 0 to $n-2^j$ in steps of $2^j$ do
        $s_{j,i} := q_{j-1,i+2^{j-1}} s_{j-1,i} + q_{j-1,i} s_{j-1,i+2^{j-1}}$
Return $s_{k,0}$
        ↓
     desired u
```

Sup $|p_i| \le b$
then

So, let us start with an algorithm for i from 0 to n minus 1. We have S 0 i equal to d i u i. Remember that, I assume the d i is then, for j from 1 to k; for i going from 0 to n minus 2 power j in steps of 2 power 0. I have to just rewrite that recurrence S of j i is q and j minus 1 i plus 2 power j minus 1 i plus q j minus 1 j minus 1 i S j minus 1 i to the power j minus 1 and return S k 0. This is our desire. You now, the time complicity is assume. Again, we will assume that each of the integer k i.

(Refer Slide Time: 16:36)



```
Algorithm
For i := 0 to n-1
    $s_{0,i} := d_i * u_i$;
For j = 1 to k do
    For i = 0 to $n-2^j$ in steps of $2^j$ do
        $s_{j,i} := q_{j-1,i+2^{j-1}} s_{j-1,i} + q_{j-1,i} s_{j-1,i+2^{j-1}}$
Return $s_{k,0}$
        ↓
     desired u
```

Sup $|p_i| \le b$
then $|q_{j-1,*}| \le 2^{j-1} * b$
$|s_{j-1,*}| \le 2^{j} * b.$

$= 2M(2^j b) * \frac{n}{2^j}$
$\le 2M(n * b)$

Total Cost
$= 2k\, M(nb)$
$= O(M(nb) * \log_2 n)$

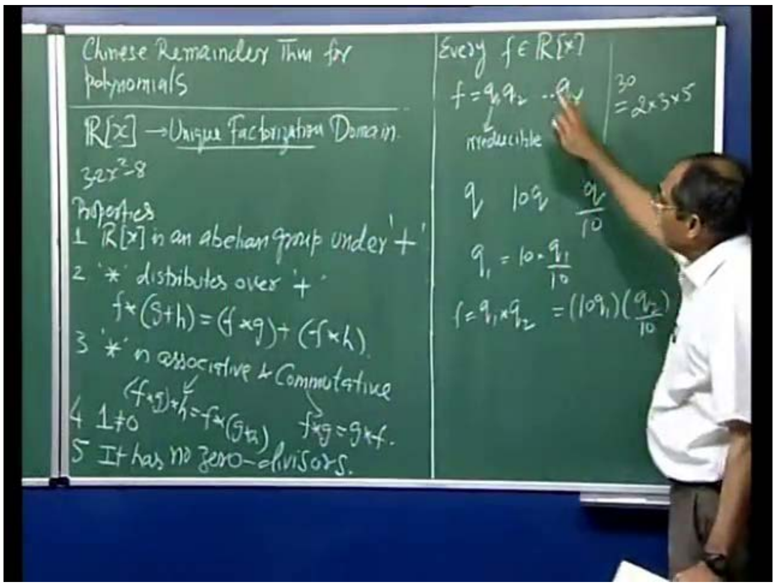
Suppose, the size in terms of bit each of the p i is less than equal to b. Then, the value of in terms of size of all the j minus 1 is less than equal to 2 power j minus 1 into b. We have to compute S j i. If the size of these no is initially, of course, what we have to do is compute d i into u i. Each of these u i is at most b bit because this is the remainder of u divided by p i. So, this is b bit because this is the remainder again. So, these are say 2 bit 2 b in size. If the size of this is 2 power j times b, if the size is towards times b, this is q power j minus one times b.

So, we have this plus, this let us see, this is actually d. So, this should add up to 2 powers less than equal to j plus 1 b bits. The reason is this plus this sum together will be at most the j plus 1 time b bits. So, what we can say is that the size of S j minus 1 star is less that equal to 2 power j times b bits. We are performing a multiplication here or 2 multiplications here. So, the time at level j for computing one of these term. We are going to perform two of these multiplications of size 2 power j b that is to say each number is less than or equal to 2 power j b bits.

There are n divided by 2 power j steps. So, we are going to perform that many sums, that which is less than equal to 2 times M n times b; once again using the fact that M is at least leaner. So, if I multiply this factor inside, that should cost at least this much; if not more. Now, this is the cost of one stage. So, I will say that the total cost is this much for each of the stages of j. They are as many as k 2 k times 2 times k times M n b which is k is log of M.

So, the cost is order M n b times log of n. So, this is a same time complexity as that of computing the forward representation, the end touple representation from given b. Now, these ideas that we have introduced for integers and the algorithms that we have given today, I am going to show can be directly. So now, let us talk about Chinese remainder theorem for polynomials.

(Refer Slide Time: 22:00)


Chinese Remainder Thm for polynomials

$\mathbb{R}[x] \rightarrow$ Unique Factorization Domain

$32x^2-8$

Properties

1 $\mathbb{R}[x]$ is an abelian group under '+'

2 '$*$' distributes over '+'

$f*(g+h) = (f*g)+(f*h)$

3 '$*$' is associative & Commutative

$(f*g)*h = f*(g*h)$ $f*g = g*f$.

4 $1 \neq 0$

5 It has no zero-divisors.

Every $f \in \mathbb{R}[x]$

$f = q_1 q_2 \cdots q_r$

irreducible

$30 = 2\times3\times5$

q $10q$ $\frac{q}{10}$

$q_1 = 10 \cdot \frac{q_1}{10}$

$f = q_1 * q_2 = (10q_1)(\frac{q_2}{10})$

Now, what we will do is first we give some background about polynomial, which justifies why these ideas can be directly lifted for them. Later on, we will come back for these algorithms. Then, finally, I will introduce special instance of p i, which is of great interest. So, let us first of all introduce certain locations and some concepts. This denotes the set of all polynomial on a single variable x. This set is something called a range. So, I am going to describe what that is unique factorization domain it satisfies. Here, R stands for number, which means the positions of the polynomial. We have representing here come from this that the fielder pre alliance and the variable.

So, for example, 3.2 minus x square minus 8 is a member of this in this range or u f. Now, the properties that satisfied are the following. One that R x is an abelian group under summation, which means if you add two polynomials, you get back a polynomial. You add f plus g or compute g plus f, you get the same thing. So, it is commutative that makes a linear for any polynomial f. There is polynomial minus f says that f plus minus f is 0. There is a 0 in it, which when added to any number gives that number back. That is known as an identity of the group minus f. f is the inverse of minus f. The second property is that we have multiplication operation, which I am denoting by star.

Sometimes, you may write as dot this operation distributes over sum. So, let me make sure I am, this operation distributes over this operation. This means f times g plus h is f times g plus f times h 3 star is associative and commutative. This says that S times g into h is equal to f times g times h. This one is saying f times g is g times f. They are equal.

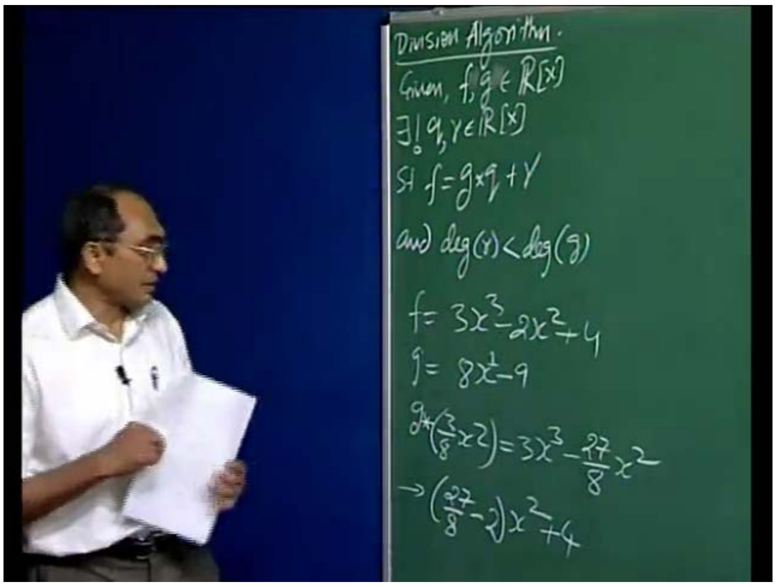
The fourth property is that one in this is a different entity 0 and something. These are the properties of a ring with a unity. That becomes a domain if you have the following property. The fifth property says it has no 0 divisor. What does that mean? This says that if you have two elements f and g in this and both of them are non zero. Then, the product cannot be 0. That is to say that the product is also non zero. Now, these properties make this a domain unique factorization. This part of the characterization of this set says that every polynomial f in R x can be factored. This can be expressed as product as in a unique way, where each of these polynomial is not factor sable or irreducible.

But, with a pinch of salt when we say that this is unique. What we are saying is that q or 10 q or q divided by 10, these are same. We do not differentiate between these because they differ by only a constant a member of the real field R, the real number field R. If I don’t differentiate between these then, I cannot factorize in any other way. The only way I can factor q 1 for that matter is equal to may be 10 times q 1 by 10. That is not much of a difference. For example, now I can say f could be written as say f has only two factors; q 1 and q 2. This could also be 10 q 1 times q 2 by 10, but this is not different from this.

So, as long as you understand that we do not consider a constant multiple as any different from the original polynomial. Then, this is unique. Now, recall that in case of integer natural numbers, every natural number can be factorized uniquely in terms of some prime. For example, we can right 30 as 2 into 3 into 5. Now, this is unique. These are not further factor sable.

They are also known as primes. In the same fashion, these are irreducible polynomial into which every polynomial can be factorized. If f was its sum irreducible polynomial, then it would be its own factor unique factor. This property is captured by unique factorization domain. Now, let us talk about few other useful properties we will need.

(Refer Slide Time: 30:57)

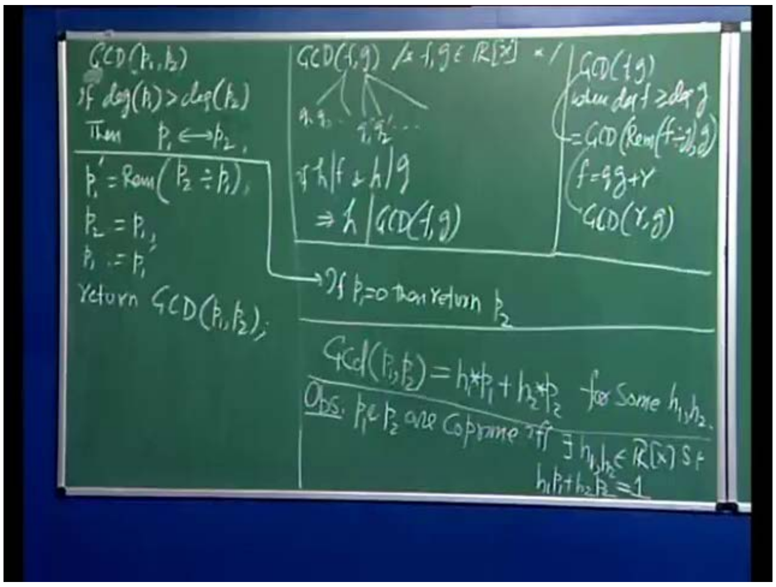

Division Algorithm.
Given, $f, g \in R[x]$
$\exists!\, q, r \in R[x]$
st $f = g \times q + r$
and $\deg(r) < \deg(g)$
$f = 3x^3 - 2x^2 + 4$
$g = 8x^1 - 9$
$g \times \left(\frac{3}{8}x^2\right) = 3x^3 - \frac{27}{8}x^2$
$\rightarrow \left(\frac{27}{8} - 2\right)x^2 + 4$

So, the division algorithm says that given f and g in R x; f and g are two polynomials. I will not bother writing down f of x g of x because it is just a bother. So, we will simplify it, f and g. Let us take two polynomials in R. Then, there exists a unique q and r also in this such that f can be expressed as g times q plus r. The degree of r is strictly less than degree of g. What we are seeing is we can divide f by g. This is the quotient and this is the remainder. When the remainder degree is less than the degree of the divisor then, there is a simple polynomial time algorithm.

To compute this, all we have to do, suppose we have, for example, 3 x cube minus 2 x square plus 4 g. Let us say, 8 x minus 9 then, I am going to multiply this by suitable monomial or term. So, that the biggest term matches with this. So, I am going to multiply g by 3 over 8 x square to get 3 x cube minus 27 by 8 x square. Now, if I divide, subtract this by this, I am going to get. I will get this minus this is 27 by 8 minus 2 x square plus 4.

Now, I will repeat this process. This we can simplify. We are going to get 11 by 8, 11 by 8 into x square. I am going to multiply it by 11 by 64 into x 11 by 64 into x. Then, subtract, keep doing it. When the degree that is the exponent of the largest term is less than the exponent of largest term, in this case, it is 1. So, it should become a constant. Then, we have this part. So, the division algorithm allows me to express every polynomial in this fashion.

(Refer Slide Time: 34:36)



GCD(p_1, p_2)

if deg(p_1) > deg(p_2)

Then $p_1 \longleftrightarrow p_2$,

$p_1' = \text{Rem}(p_2 \div p_1)$,

$p_2 = p_1$,

$p_1 := p_1'$

return GCD(p_1, p_2);

If $p_1 = 0$ then return p_2

GCD(f, g) /* $f, g \in \mathbb{R}[x]$ */

$q_1, q_2, \ldots \quad q_1', q_2', \ldots$

if $h \mid f$ & $h \mid g$

$\Rightarrow h \mid \text{GCD}(f, g)$

GCD(f, g)

when deg f ≥ deg g

$= \text{GCD}(\text{Rem}(f \div g), g)$

$f = qg + r$

GCD(r, g)

$\text{Gcd}(p_1, p_2) = h_1 * p_1 + h_2 * p_2$ for Some h_1, h_2.

Obs. p_1 & p_2 are Coprime iff $\exists\, h_1, h_2 \in \mathbb{R}[x]$ s.t.

$h_1 p_1 + h_2 p_2 = 1$

Given g, the next thing is competition GCD, the existence of GCD. So, suppose we have f and g. f and g are from R x. Then, there is a greatest common divisor of these two polynomials, which is a polynomial. Also x; by definition that means that that divides both f and g. Anything that divides both this polynomial divides that. So, if x divides f and g then, this polynomial will divide x. We will not bother to prove this. But, I can just point out the existence. This is easy to see because if I compute the factors of f and g. Suppose, I break the factors q 1, q 2.

The factors of this q 1 prime, q 2 prime. Notice that they may differ. I am saying q 1 is equal to q 1 prime. If they differ by a constant so, constant can be this. Then, we find out all the common factors. If I collect them and multiply together, this would be the GCD. It is easy to see that. So, the existence of GCD is easy to see in any unique factorization domain.

Next, I am going to show how to compute this. So, in mean time, I just point it out. If h divides h, h divides f and h divides g that would imply that h divides GCD. I think I made a mistake. Earlier I said GCD divides h but no, this is the greatest common divisor. Hence, h will divide this as well. Now, the computation of the GCD comes from the simple fact, which is identical to algorithm applicable integers applies here as well. That is, to say that GCD of f, g where the degree of a f is greater than equal to degree of g.

If this is the case then, this is also equal to the GCD of the remainder of f divided by g, g. Notice that, we earlier showed f can be written as q g plus r. So, this is GCD of r, g. Now, anything that divides both of them, that will divide the right side of this expression. It divides r and g. Hence, it will have to divide f. Now, anything divides both of these that would divide f as well as g. r can be written as f minus q g; hence, it divides r.

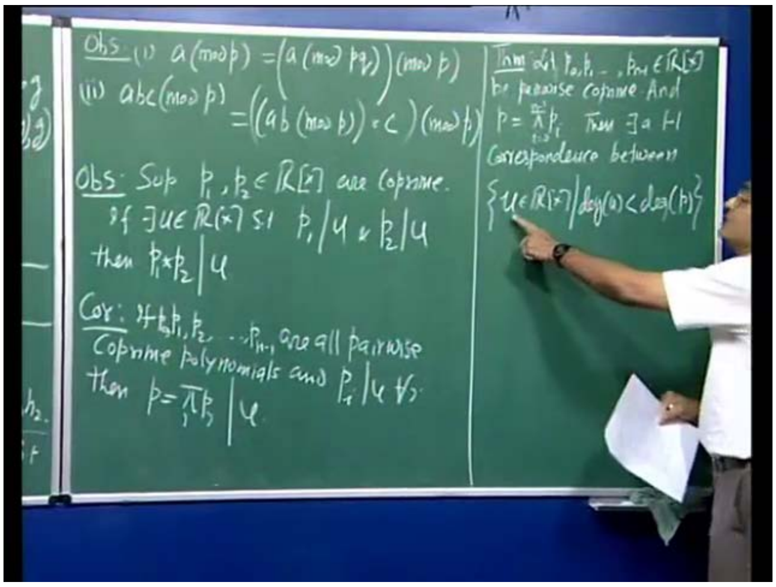
So, notice that the degree of this is less than that of g. So, I can alternatively go down. Finally, I will have one of these terms as 0. GCD of r, 0 is r because g divides r, r divides g. So, the GCD computation, GCD of p 1, p 2 is if degree of p 1 is greater than degree of p 2 then, exchange p 1 and p 2, then I will compute p 1 prime as the remainder of p 2 divided by p 1. I will define now p 2 as p 1 and p 1 as p 1 prime.

Once again, the degree of this is greater than equal to degree of this. Then, return the GCD of p 1, p 2. The terminating condition is that if here, I am going to say if p 1 is 0 then, return p 2. So, terminating condition some here, we know that degree of p 1 is greater than or equal to the degree of p 1. In case p 1 is 0 then, you return p else you precede this. Now, this is exactly what we do in the case of the integers. Hence, we also have one interesting property. That is GCD of p 1 and p 2 can be expressed as some h 1 times p 1 plus h 2 times p 2.

That is simple because in the base case GCD of r, 0 is r. So, the GCD is expressed as 1 time star. That is done. Then, you can express this as linear combination of r and g. It means that you multiply a polynomial here, a multiplier polynomial here, and the sum transferred to the GCD. Then, I can then replace r by f minus q g. I will get a linear combination in terms of f and g.

So, I will get GCD of f and g in terms. So, this is very useful result, this tells me that so observes that that p 1 and p 2 are coprime; if this one, I should say for some h 1 and h 2. This says their coprime remember, coprime means the GCD is 1. So, this is h 1 h 1 and h 2 in r z r x such that h 1 p 1 plus h 2 p 2 is 1 if you can combine the get one is their coprime. Two things about modulo computation are given.

(Refer Slide Time: 42:56)



Obs: (i) $a \pmod{p} = \left(a \pmod{pq}\right) \pmod{p}$

(ii) $abc \pmod{p} = \left(\left(ab \pmod{p}\right) * c\right) \pmod{p}$

Obs: Sup $p_1, p_2 \in \mathbb{R}[x]$ are coprime. If $\exists u \in \mathbb{R}[x]$ s.t. $p_1 | u$ & $p_2 | u$ then $p_1 * p_2 | u$

Cor: If $p_0, p_1, p_2, \ldots, p_{n-1}$ are all pairwise coprime polynomials and $p_i | u \; \forall i$ then $p = \prod_i p_i \, | \, u$

Thm: Let $p_0, p_1, \ldots, p_{n-1} \in \mathbb{R}[x]$ be pairwise coprime And $p = \prod_{i=0}^{n-1} p_i$. Then $\exists$ a 1-1 Correspondence between $\{u \in \mathbb{R}[x] \,|\, \deg(u) < \deg(p)\}$

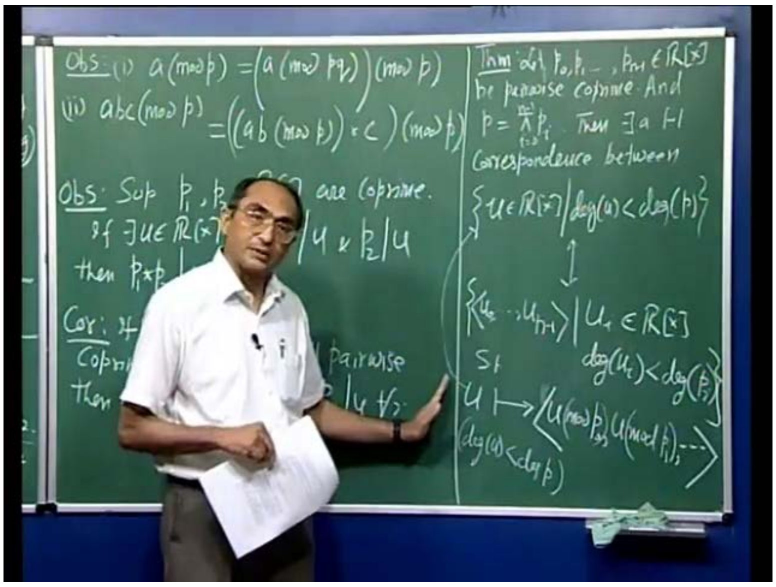
So, few observations which says that a modulo b, this is we have seen in the space of integer a modulo p times q modulo p. Similarly, a times b times c modulo p is equal to a times b mod b times c mod p. So, these are very simple statements, easy to verify. Then, we have another observation. It says p 1 and p 2 are coprime. Suppose, p 1 and p 2 are integers and x are coprime. If there exists u in R x such that p 1 divides u. Notice that, when you divide u by p 1 and the remainder is 0 u p 1 divides u. Similarly, if p 2 divides u then, p 1 times p 2 divides u.

Again, if you recall the fact that there is a unique factorization p 1 p 2 and u then, this says the factors of p 1 are all present in u; this same with this greater than equal to the multiplicity of u. Suppose, I have got q 1here, q 1 power 2 here. Then, we have q 1 power 2 or more and so on. The factors of this are also present in this. But, the factors of p 1 and p 2 share nothing because they are coprime.

Hence, both combine factors p 1 and p 2 must also be present in this. Product multi divides the entire u. The coronary of this statement is that if we have p 1 p 2. I would say p 0 p 1 p 2 p n minus 1 are all pair wise coprime polynomial. p i divides u i, for all i just extending. The same result we can say then, p which is the product p i divides u the entire product will divide u. Now, we are going to state the Chinese remainder theorem in this way. This statement is exactly the same. The theorem says let p 0 p 1 p n minus 1 belonging to R x be the entire pair wise coprime, p be the product of p i 0 to n minus 1.

Now, we will again, we have a set, 2 sets here. We are going to establish it correspondingly. So, the two sets are the following. Here, we have set of one. Then, there exists a one to one correspondence between the set. Here is all the polynomial u in R x, where the degree of u is less than the degree of b. Earlier we had integers and the integers were ranging from 0 to p minus 1. Here, we are talking about similar thing but in terms of the degree. The degree is up to 1 less than the degree of p this and this set of the other side, is a touple u 0, 2 u.

(Refer Slide Time: 49:00)



Obs: (i) $a \pmod p = (a \pmod{pq}) \pmod p$
(ii) $abc \pmod p = ((ab \pmod p) \cdot c) \pmod p$
Thm: Let $p_0, p_1, \ldots, p_{n-1} \in \mathbb{R}[x]$ be pairwise coprime. And $p = \prod_{i=0}^{n-1} p_i$. Then $\exists$ a 1-1 correspondence between
$\{u \in \mathbb{R}[x] \mid \deg(u) < \deg(p)\}$
$\{\langle u_0, \ldots, u_{n-1} \rangle \mid u_i \in \mathbb{R}[x]$ st $\deg(u_i) < \deg(p_i)\}$
$u \mapsto \langle u \pmod{p_0}, u \pmod{p_1}, \ldots \rangle$
$(\deg(u) < \deg p)$

Sorry, n minus 1, where each u i is a member of R x and u i is the remainder of u divided by p i. Notice that, the remainder has degree strictly less than that of the divisor. So, you do not need to say this. Here, we are simply saying the sets say, that the degree of u i is strictly less than the degree of p i. Now, this defines the set of end touples of polynomial, where the highest polynomial degree less than the degree of p i. Now, these two sets have one to one correspondence such that u in this set maps to u modulo p 0 u modulo p 1 to this touple. These have degree less than the corresponding p i. This has degree less than p 0. This has degree less than p 1.

So, this touple falls in this set belongs to this set. This means that to say the degree of u is given to be the degree less than of p so that the similarity is there. The only difference is that we will be looking at the degree otherwise there is a identical correspondence. Once again, we can compute this touple from this polynomial and this polynomial from this touple of polynomial in the identical way that we have computed in the integers. So, I will go through that in the next lecture. Then, we will consider a special case of p i. We will see some surprising consequences of correspondence and these algorithms, which are very useful in the next class.