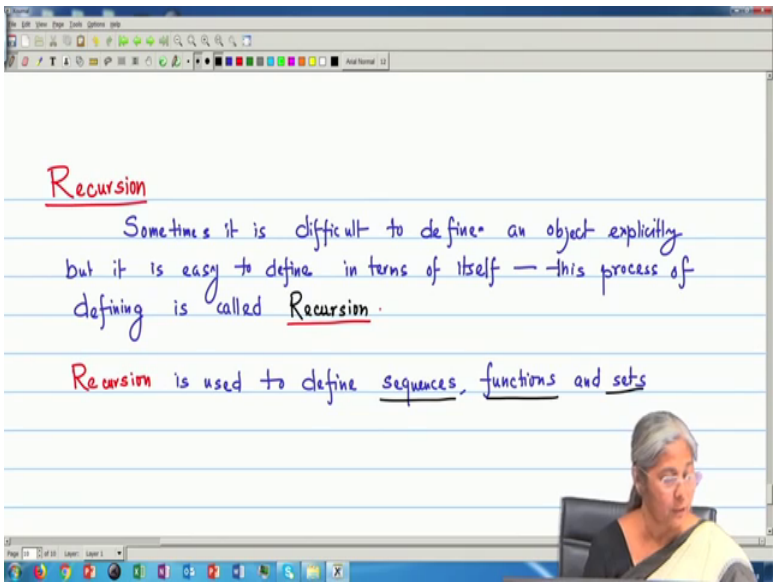
Discrete Structures
Prof. Dipanwita Roychoudhury
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 26
Recursion

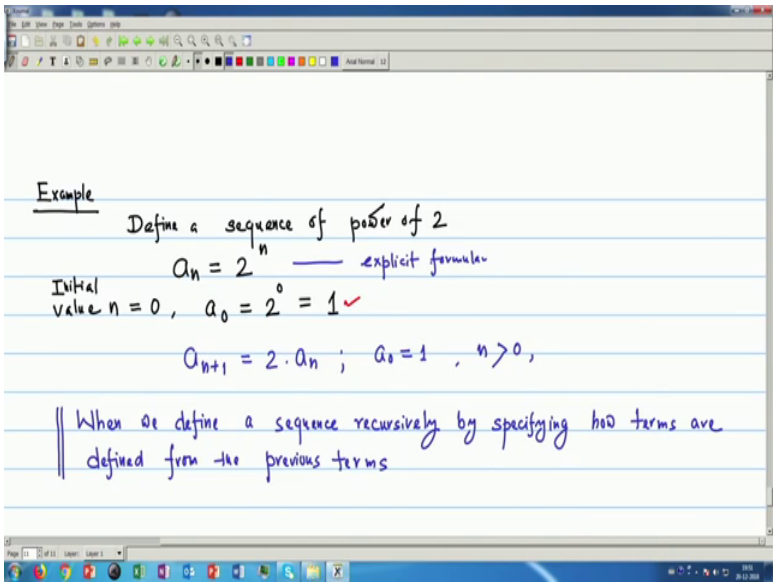
Today we will learn a very important concept of computer science called the Recursion. And first the very simple concept that what we concept of recursion, that what we mean by recursion. And how actually it is we define recursively the function, set, sequences today we will read that thing. And, slowly then we will be going to that how recursion is used to define relation, how to solve etcetera. So, today we will read that recursion. Sometimes it is difficult to define an object explicitly.

(Refer Slide Time: 01:30)



So, you can write that sometimes it is difficult to define an object explicitly. But it is easy to define in terms of itself and these process of defining the object in terms of itself is called recursion. So, very simple way if I took that this process of defining is called recursion. Now, we recursion normally we use to define the sequences, the functions, the sets and later we will use for many real life problems. So, you can write is to define sequences, functions and sets. These are the three basic things we will see that how recursion is used. Now, we take one simple example to illustrate the concept of this recursion.

(Refer Slide Time: 04:30)



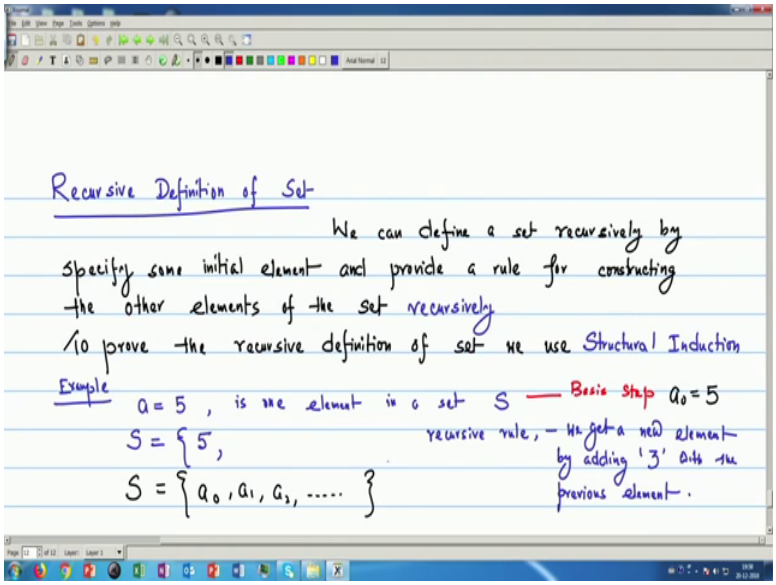
We take one example ok, we define a sequence of power of 2. How I write? Say I give a sequence n equal to 2 to the power n. Now, these same sequence since it is power of 2 I can define in a different way, but I need some initial value. So, what is initial value? Say for n equal to 0 I say I have some initial value say n equal to 0, we know a 0 is 2 to the power 0 is 1. So, if this is given then I can write the same sequence as say n plus 1 is 2 a n where, a 0 equal to; 1 and this is defined for n greater than equal to or n greater than 0; that means, n equal to 1 2 3 like that ok.

Now, see I tell that this is some explicit formula or some expression, explicit formula for representing a sequence of power of 2, but this is same if I write n plus 1 equal to 2 a n. Now, how recursion we have defined that it is a process of defining itself; that means, only as if the next term n plus 1 we are getting from a n and we are writing that as a formula. So, if we remember mathematical induction this is very similar to that thing because, induction also we have tried. How one particular term can be represented using the previous terms either one term or some previous few terms.

So, this induction and recursion that has a relationship and that we will explain now. So, when we define a sequence recursively so, we write the basic thing that when we define a sequence recursively. The most important thing is that we have to find out or we have to specify that how one particular term, how terms are defined from the previous terms and see this is same as that of a what is we have called the mathematical induction also

we have used. So, to prove normally we use mathematical induction for these recursive definitions of sequences. Now for sets this is for sequences.

(Refer Slide Time: 10:05)



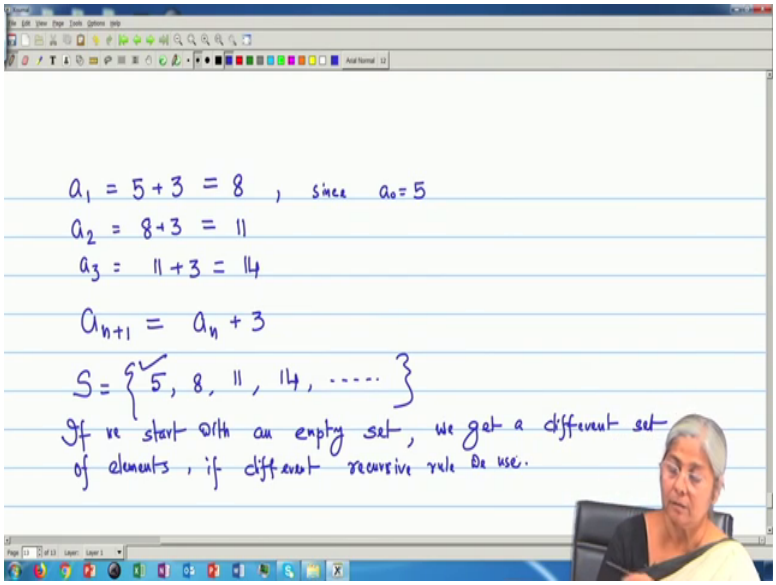
So, if we define sets recursively, recursive definition of set. Here also see we need some initial element of a set and then we give a relation that how other elements of the set can be added or how they are related and in that way we can generate the full set. So, we can write that we can define a set recursively by specifying some initial element and then provide a rule and for constructing the other elements of the set. And, this rule actually this rule must be or some recursive way we can define. So, we can define recursively, you can just write rule for constructing a set recursively.

Now, here also to prove the recursive definition of set we can use we use structural induction. Since this is a construction; construction of the set so, we use structural induction. Give one example say as if one element 5 is there so, I get one element a is 5 in a set 5 is one element in a set S. So, my initially my as if my S I consider that 5. We can think as if this is my basis step if I compare from the mathematical induction as the I can think that this is the basis step. For if I consider my S is say S is a 0 a 1 a 2 like this are the elements then I can think that basis step as if the for a 0 equal to 5 ok.

Now, we told that one rule recursively I can tell some rule very simple rule if I consider what is that recursive rule. That recursive rue is I can tell that recursive rule that each element is again we get the new element by adding 3 from the, with the previous

elements. So, as if the next element recursive rule is the each element or we get the next element we get we get a new element by adding 3 with the previous element.

(Refer Slide Time: 17:06)

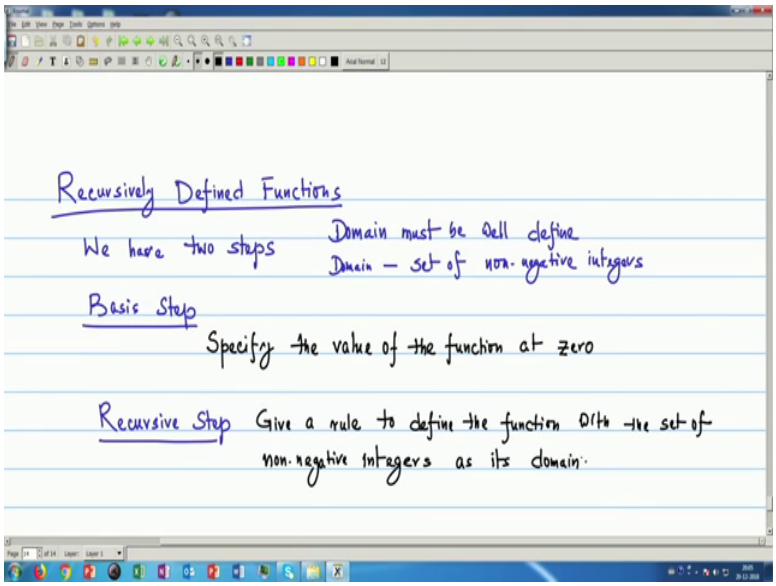


So, what is my a 1? So, I get a 1 is 5 plus 3 because, a 0 was 5 since the first element was defined a 0 was 5. So, I get the next element of the set is 8 plus 3 is 11, next element I got always if the previous element I add and this is I am telling that recursively I get; that means, I can write sum n plus 1th element I can get with adding 3 with the previous element.

So, finally, I got my set as if I start with 5 then 8 then 11 14 like that it will be we will get in this way. Now; obviously, I could start from a empty set. That I then in that case it will be a if I would start with the empty set; that means, if we start we will get the same thing. So, I started with as if this is my initial element or I could start with an empty set we get the same set if, but rules will be different we start if we start with empty set our rule may be we give the same set with different rule.

We get a different set of elements if different rule is different recursive rule we use ok. Now, we see recursive functions because that is most important that we can define the recursively define functions.

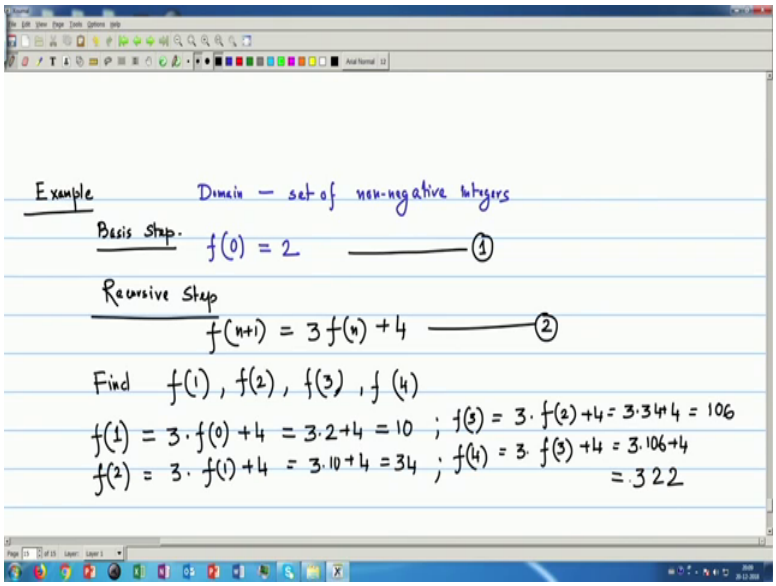
(Refer Slide Time: 20:46)



Now, here always we have two steps when we define recursive function we have two steps this is very similar to mathematical induction, we give a basis step where we specify where we specify the value of the function. So, I am considering at 0, then the recursively defined function must be must have some well defined domain; domain since it is a function the domain must be well defined and say here I am considering the domain is set of non-negative integers.

So, I am considering as if 0 at 0 I am specify the value ok. Now, the recursive step now I have recursive step, this is very similar to our inductive step that we give a rule to define the function with the set of the non-negative integers as the domain. So, my function is such or my rule is recursive rule is such it should not violate the domain that we have defined in the problem as its domain this is very important thing and we must remember. We take a very simple example of recursively defined function.

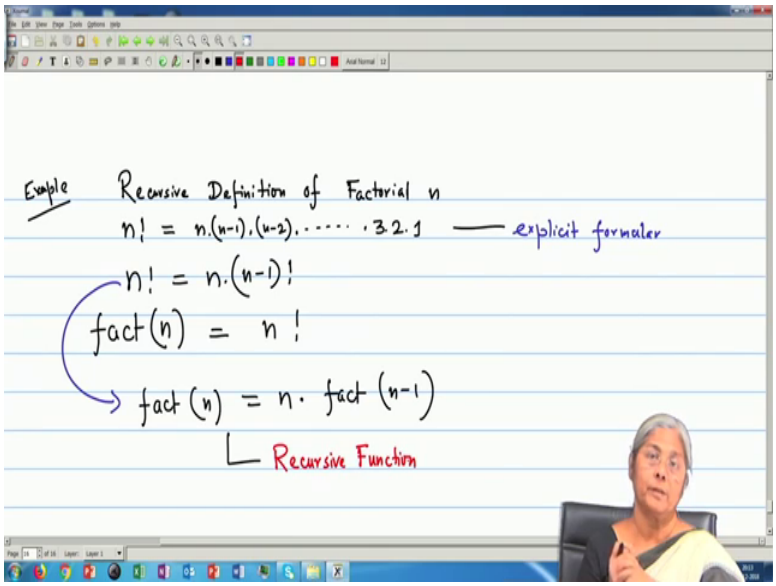
(Refer Slide Time: 24:26)



Say basis step; so, we take simple say f 0 equal to 2, here also I am considering for these example the domain is set of non negative integers. So, f 0 equal to 2 and the rule the inductive or recursive step say we define something f n f of n plus 1 is something 3 of fn plus 4 ok. So, now we can find we can find f of 1 f of 2 like f 3 f 4 and so on so, up to this. So, how recursively we can do this function we can define. So, f 1 is we can use the relation that say this is my basis step and this is my recursive step. So, f 1 is 3 f 0 plus 4 is equal to 3 into 2 plus 4 equal to 10 similarly what is f 2 f 2 is 3 f 1 plus 4 is 3 into 10 plus 4.

Similarly, I can do f of 3 is f 2 plus 4 s 3 into 34 plus 2 is 104 then f 4 is 3 into f 3 plus 4 s 3 into 1 oh sorry this should be 4, now 2 because my recursive step is 3 fn plus 4 it will be 6. So, we recursively the definition that we have given that it is defined by itself now we see that every previous term; that means, if the function the same function we are using only for the previous term and this is very similar to our mathematical induction. Now, we see that very simple example that all of you know that factorial in how to compute factorial n.

(Refer Slide Time: 29:04)



So, another example we see recursive definition of factorial n we know that n factorial is n into n minus 1 n minus 2 its 3 2 1. So, this is my explicit formula this is my explicit formula for factorial n. Now, we can easily define this thing that n factorial is n into n minus 1 factorial. So now, if I define a function that say fact that fact n is nothing, but n factorial. So, I can write this thing as fact n equal to any two fact n minus 1. See that same function only with different arguments that is used to define a recursive function and this concept is actually our how recursively defined function we use here.

So, this is the concept of or this is the way we define the recursive function. And in computer programming this is a very important role. We will read in the next class, there are many examples or many problems that are much easier way we can define with recursion. And, sometimes when we write program the computer programs then recursion is very important role. We can very easily we can write recursive program rather than or call iterative program, that we will see in the next classes.