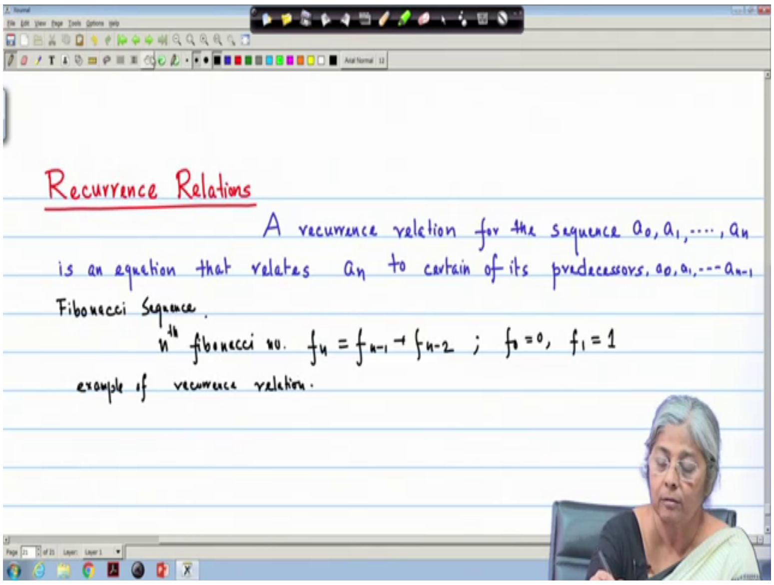
**Discrete Structures**

**Prof. Dipanwita Roychoudhury**
**Department of Computer Science and Engineering**
**Indian Institute of Technology, Kharagpur**

**Lecture – 31**
**Recurrence Relations**

So, we have read the recursion and the recursively defined sets, functions, sequences and we have also learned that how to design the recursive algorithms. Today will read the Recurrence Relation, which is very much related to this recursion. And will see that, how given a sequence that each term is related to others and how we can get a explicit formula, how we can get a recurrence relation with that thing and how to solve or what do you mean by solution. So, will do that framing recurrence relation and solving recurrence relations will read the recurrence relations.

(Refer Slide Time: 01:20)

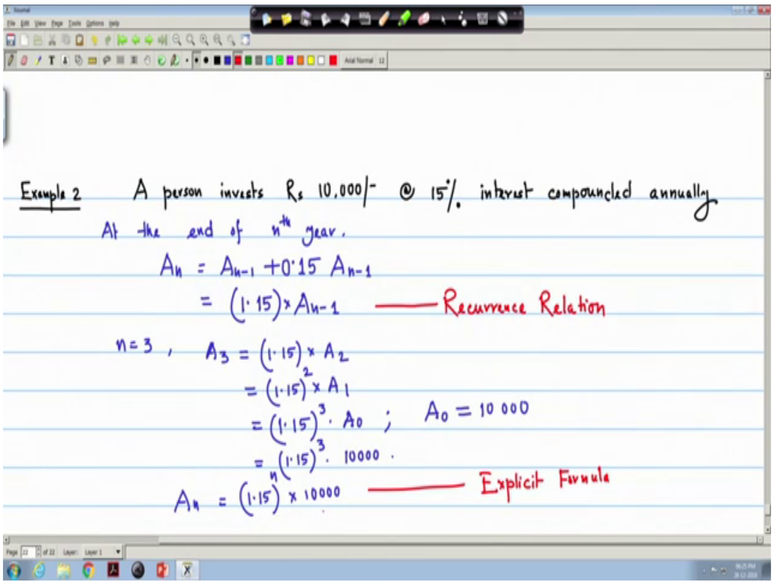



So, first we define a recurrence relation, I can write that a recurrence relation for the sequence, we have earlier read the sequence or recursively defined sequences. So, for the consequence a 0, a 1 to a n having n number of sequence. It is an equation that relates a n to certain of its predecessors and the predecessors are a 0, a 1 up to a n minus 1.

So, recurrence relation gives a relation that how a n is related to its predecessors. Now we have already read many sequences where this relation exists. So, one popular example is that Fibonacci sequence, how we define that Fibonacci. So, for Fibonacci sequence we know that nth Fibonacci number f n nth Fibonacci number is a n or f n f n is f n minus 1 plus f n minus 2 and some initial conditions are given that f 0 equal to 0 and f 1 equal to 1.

So, this is an example of a recurrence relation because, f n is related to its predecessors f n minus 1 and f n minus 2 with some addition and these are my initial conditions. So, this is an example of recurrence relation, we taken example of compound interest, how to compute the compound interest, we take one example.

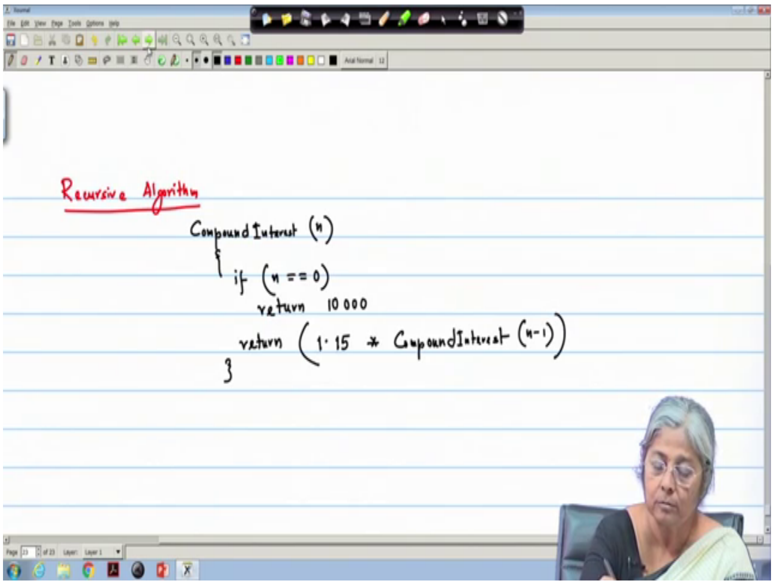
(Refer Slide Time: 05:17)




Example 2, see person invests rupees 10000 at the rate of say 15 percent interest, which is compounded annually. Now we want to find out the total amount at the end of nth year. So, we can write the amount that the at the end of nth year, we can get that A n is A n minus 1 that amount that at the end of n minus 1 th year plus the interest; that means, it is 15 percent. So,15 0.15 into A n minus 1. So, I can write this is 1.15 into A n minus 1. So, here A n relates to only 1 predecessors A n minus 1 in this wait 1.15 into A n minus 1. So, this is a recurrence relation, this is a recurrence relation.

Now, if I take that n equal to 3 see n equal 3. So, A 3 is 1.15 into A 2, I can write 1.15 into again it is 1.15 we want. So, this is square A 1 is 1.15 cube into A 0. Now what is my A 0? Because at the beginning it in 10 rupees 10000 was invested; so, it is 10000. So, I can write that it is 1.15 cube into 10000. So, I can generalize this thing as that A n equal to 1.15 whole to the power n into 10 10000 and we call this is a explicit formula, this is an explicit formula or of n. So, this is my recurrence relation, where the nth term, how it is it gives a relation, how it is related to the predecessors and it gives directly; that means, it is a function of n explicit formula. Now, we have read the recursive algorithm also.

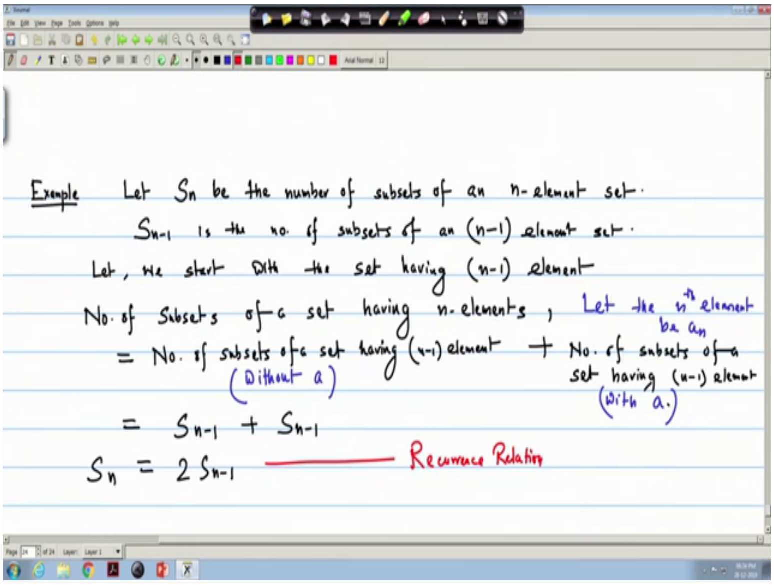
(Refer Slide Time: 09:48)




So, the same for the same example, I can write the recursive algorithm that, I can write the function that if it is compound interest n; that means, at the end of nth year. Now if n equal to 0; that means, it is my initial condition or the terminating step then return 10000 else it will return that compound interest delete 1.15 into compound interest n minus 1.

So, with this simple example, you see that what that we get the recurrence relation explicit formula and this recurrence relation; that means, the recursive step that if we give a code of that then it gives you as the recursive algorithm on that thing. Now, we take another simple example, but it is very important when will see the will try to solve the recurrence relation.

(Refer Slide Time: 11:50)

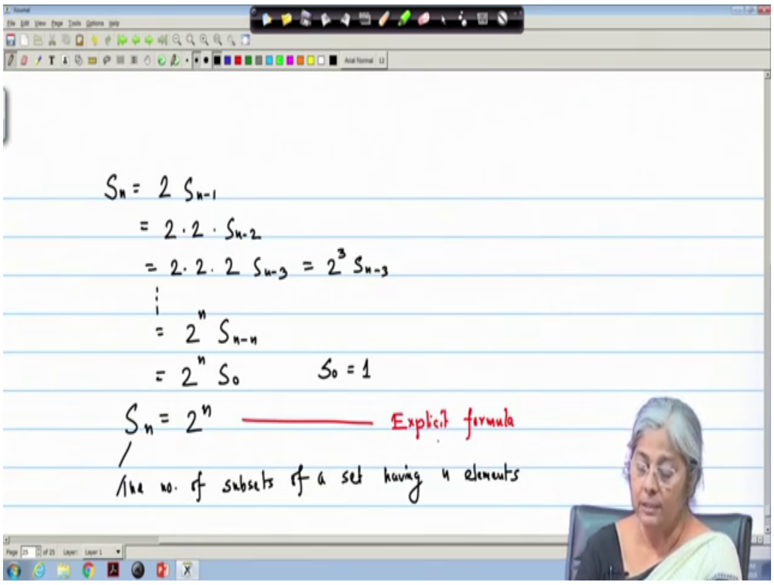



So, earlier we have read one example of set that say let S n be the number of subsets of an n element set then how can I give a recurrence relation for a set, if we remember that see S n be the number of subset then S n minus 1 is the number of subsets of an n minus 1 element set.

So, let initially I have let we start with the set, we start with the set having n minus 1 element. Now if I get 1 element to that set. So, that it becomes an n element set then what will be the number of subsets? So, the number of subsets or of a set having n elements, I can write this equal to the number of subsets having n minus 1 elements subsets of a set having n minus 1 element plus, if I add one more element to all the subsets of this n minus 1 elements; that means, this is if I let the nth element be let the nth element be a or a n.

So, number of subset of a set having n minus 1 element, which is without a and here I can write the number of subsets of a set having n minus 1 element with a, this is with a. Now clearly this is my S n minus 1 and this is also S n minus 1 only difference is here, in these subsets a was not there and in this subsets S is there. So, this becomes 2 S n minus 1. So, S n equal to 2 S n minus 1, this is my recurrence relation, this is my recurrence relation.

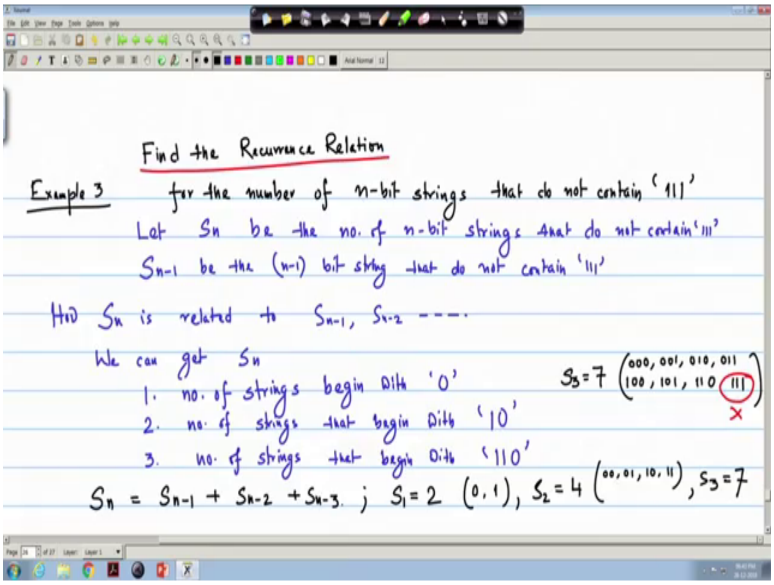
(Refer Slide Time: 17:23)




Now, we can get the explicit formula also for this example, if I start with that S n equal to 2 S n minus 1 is 2 into 2 S n minus 2 is 2 into 2 into 2 into S n minus 3. In that way, if I continue I will be getting that 2 to the power n, which is here 2 cube into S n minus 3, this is actually 2 cube into S n minus 3. So, I can get 2 to the power n S n minus n is 2 to the power n is 0 and or S 0 is 1, which is the empty set, this is 2 to the power n.

So, S n is 2 to the power n and this is my explicit formula, for the number of subsets S n is the S n is the number of subsets of a set having n elements. So, we see that how we can frame the recurrence relation and we get a explicit formula actually, this explicit formula is called the solution and the solution that which satisfies the sequences, we get. Now we get another example we see.

(Refer Slide Time: 19:19)




So, we find the number of the bit n bit strings, find the number of n bit strings that do not contain 1 1 1 say, a particular bit pattern is not there then how to solve or how to get this number. So, let the solution is let S n be the number of n bit strings that do not contain 1 1 1 pattern. Now we want to find out the recurrence relations actually here, that example will be that find the find the recurrence relation, because first we try to frame the recurrence relation, how to frame the recurrence relation.

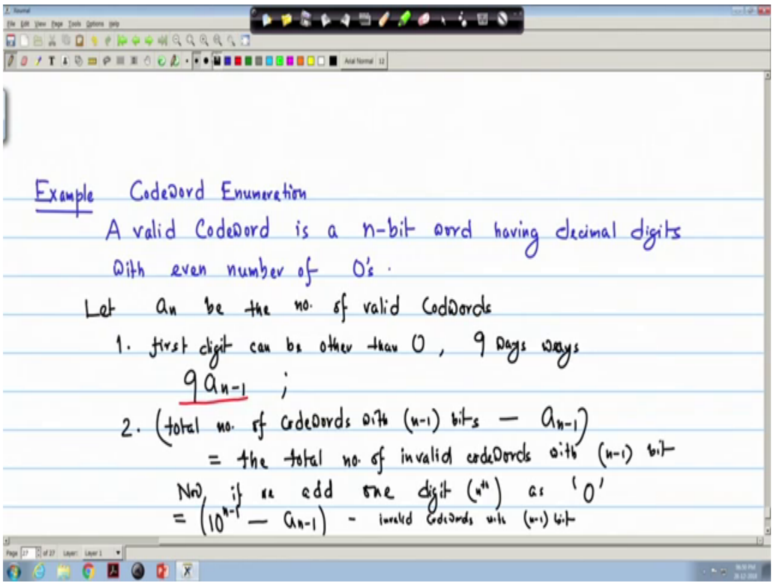
So, our problem is find the recurrence relation, where for the number of n bit strings, but we write for the number of n bits string that do not contain 1 numbers. Now let S n be the number of n bit strings then; obviously, S n minus 1 be the n minus 1 bit string that do not contain 1 1 1. And then similar thing, where is n minus 2 is n minus 3 like that then how we can get because, recurrence relations we have defined that a particular term, how it is related to it is predecessors; that means, here how S n how S n is related to it is predecessors bit S n minus 1, S n minus 2 and so on.

Now, how we can find S n because, it should not contain 1 1 1. So, I can get S n. So, I can get S n that will found that that it is the strings the number of strings that begin with 0, number of strings that begin with 0 because, it they do not contain 1 1 1. So, if it begins with 0. So, then it will not contain the 1 pattern then number of strings that begin with 1 0 then also it is violated number 3 is the number of strings that begin with 1 1 0 says that 1 1 1, it does not consider.

So, I can write that expression that S n, I can write S n equal to S n minus 1. For the first case, then S n minus 2 plus S n minus 3 then what will be my initial conditions? My initial conditions will be S 1 equal to 2 since, only 1 bit S 1 that it will it can be only 0 and 1. So, it is 2 then what will be my S 2? S 2 will be 4, because these will be 0 0, 0 1, 1 0, 1 1. Similarly, I can get S 3 equals to 11, I can get S 3 equal to 7 because here, it will be 0 0 0, 0 0 1, 0 1 0, 0 1 1, 1 0 0, 1 0 1 and 1 1 0, but 1 1 1 will not be here, these will not be here because so, it has 7.

So, the initial conditions that S 1 equal to 2 S 2 equal to 4 and S 3 equal to 7, I get the recurrence relation that is n equal to S n minus 1 plus S n minus 2 plus S n minus 3. So, I can frame the recurrence relation for S n, which is the which represent the n bit string that do not contain 1 1 1. Now we can see that a different type of examples normally, it is very important that we call that a codeword enumeration.

(Refer Slide Time: 27:15)

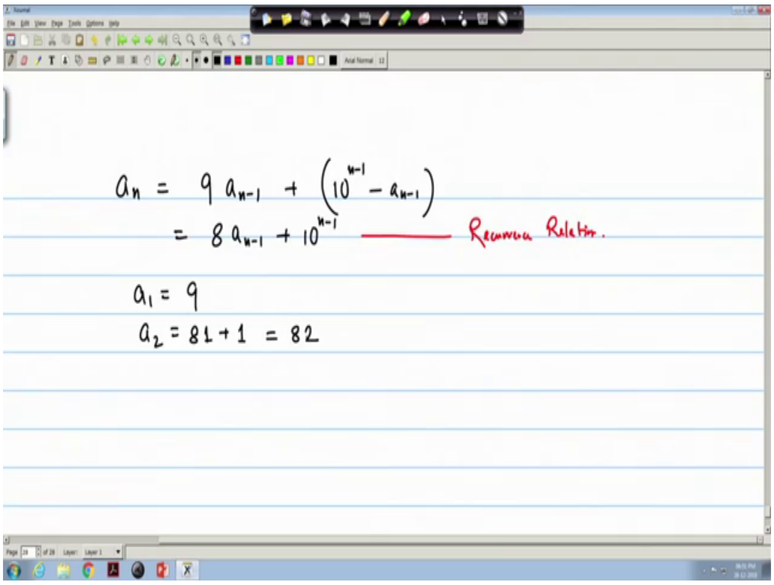



So, we take another example of codeword enumeration. Say a code a valid codeword so, a valid codeword is a some n bit word having decimal digits with even number of 0's. So, it is defined like even number of 0's, now we have to find the codeword. So, let a n be the number of valid codewords then, how we can get a relation or a recurrence relation? So, I can write that a n and here also I can get set, how we can get the recurrence relation of the n or how we can found the number of valid codewords? So, it can be the first word, first letter or the first digit can be other than 0. So, there are 9 such digit, digits can be there.

So, 9 ways I can do that thing, we can do and how many such we can get. So, this going to be 9 a n minus 1, because n minus 1 is a again a valid code word; that means, we consider n minus 1 is the n minus 1 bit words having even number of 0's and other way other things, we can get that the total number of codewords. So, if I can add a 0 total number of code words having n minus 1 bits. So, total number of codewords with n minus 1 bits minus a n minus 1, the valid codeword. So, it will give me the total number of it will give me the total number of invalid codewords, the total number of invalid codewords with n 1 minus bit.

Now, if we add 1 digit the nth digit say the nth digit as 0 then I will get a valid codeword, I will get a valid codeword. So, I can tell that these nine a n minus 1 and these total number, what will be that number? We can we can see that, these number total number will be the actually give me the n. Now, what is this number? This number will be we know that since, I have 10 digits. So, these will be 10 to the power n minus 1 is the total number of codewords minus valid codewords is n minus 1. So, this is number of a invalid codewords, this is invalid code words with n minus 1 bit and n th bit we add a 0.

(Refer Slide Time: 33:49)




So, n becomes a n is 9 a n minus 1 plus 10 to the power n minus 1 minus a n minus 1 and this gives me 8 a n minus 1 plus 10 to the power n minus 1, this is my recurrence relation then what is the initial condition? This is my recurrence relation. We can get that for a 1 the initial condition is that a 1 equal to 9 and a 2 equal to some 81 plus. So, 10 to the power 0 is 1 that we can validate that it is equal to 82. So, you see a different type of examples and how we can frame or how we can design the recurrence relation for those problems; that means, given sequence that how a 1 term is related to the previous terms. So, next class we will see, how it can solve the recurrence relations.