INDIAN INSTITUTE OF TECHNOLOGY MADRAS

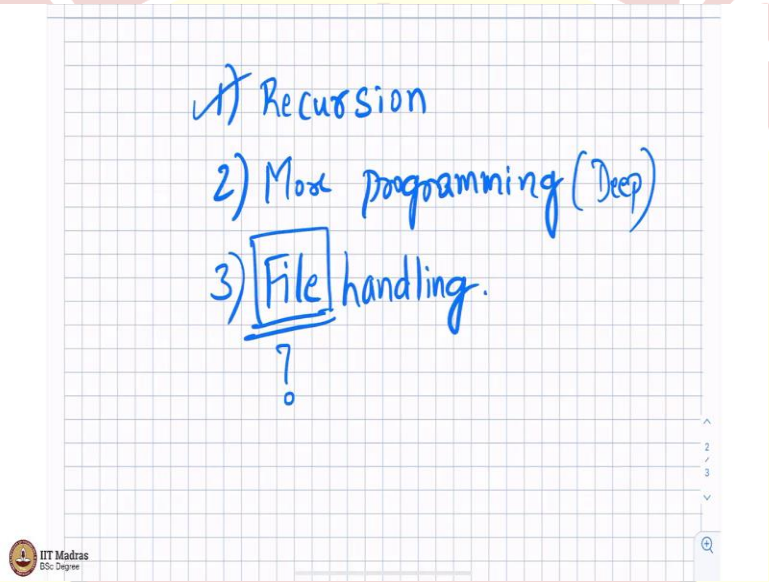
सिद्धिर्भवति कर्मजा

IIT Madras

ONLINE DEGREE

Programming in Python
Professor Sudarshan Iyengar
Department of Computer Science & Engineering,
Indian Institute of Technology Ropar
Mr. Omkar Joshi
Course instructor
Indian Institute of Technology Madras
Online Degree Programme
Introduction to the week and introduction to recursion

(Refer Slide Time: 00:16)



1) Recursion
2) More programming (Deep)
3) File handling.
?

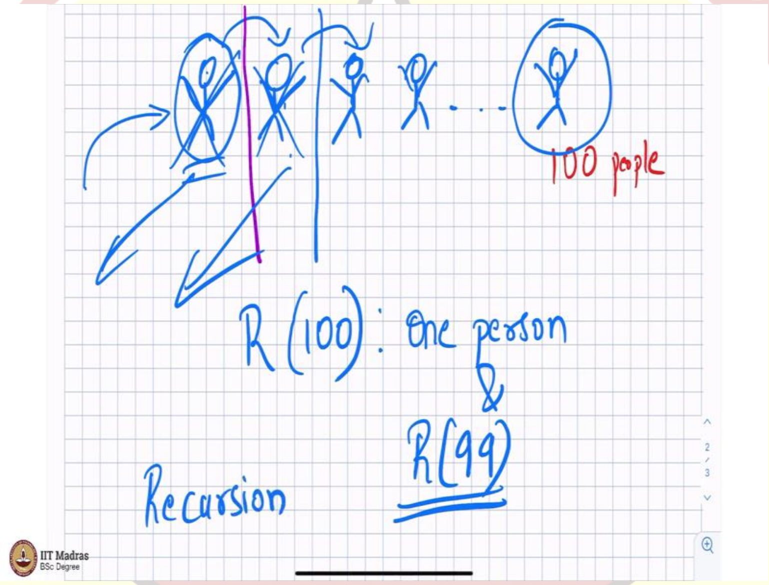
So, this week we will be seeing more of recursion simply because many of you have been asking for it. I saw this over the forum and then the general discussions I have had with some students. They said, they want more of recursion. So, anyways, that was in line. So, we are going to do more of recursion and we will do more of programming, because that is why we are here.

As you see, Python is about programming, where car is for driving, bicycle is for riding. So, Python is all about programming. There is no point in just going ahead with more and more theory. So, we will do more of programming. And mostly, it will be some deep diving this time. So, we may discuss a few non-trivial things.

And finally, the third and the last point that we will be discussing this week is some file handling. May sound technical for people who would not have heard of file handling, what exactly one means by file handling. We will start from the scratch and make you feel comfortable with how to handle data if it is really big.

You see even lists have some disadvantages. You cannot go beyond the limit. But with files on your hard disk, you see you store a movie on your hard disk. You store your MP3 songs on your hard disk. So, you can also store big, big, big files on your hard disk and how do you access them and how do you make sense out of it. That is what we will be discussing in file handling. So, let us get on with recursion right now.

(Refer Slide Time: 01:48)

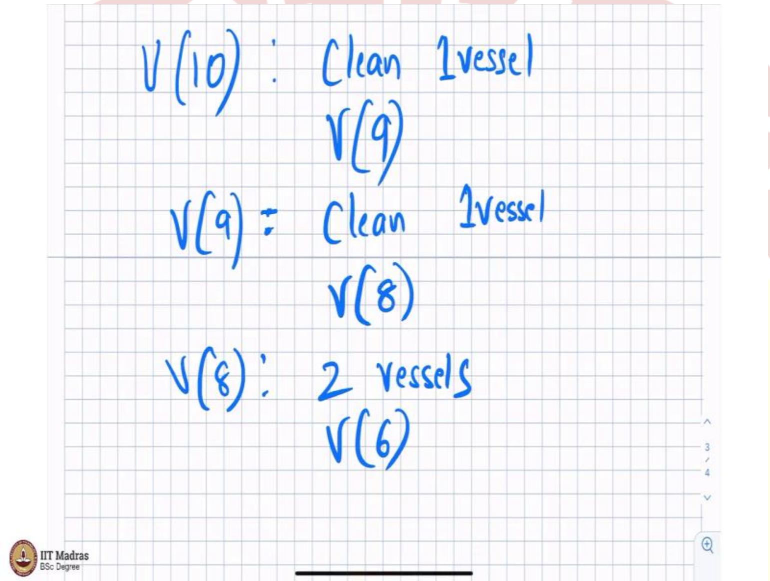


So, what is recursion? So, we have been thinking about recursion from the past two weeks. And I am sure most of you are now mature enough to understand what is recursion, but I will go with more examples just so that we feel even more comfortable. So, assume there are some n people in queue, 1, 2, 3, 4, and so on. There are some, let us say, 100 people in queue.

So, what I want to do is I want to sneak in here and then spread a piece of gossip. But I will tell this person and just tell him to tell to his neighbor and then forget it, and then move on. He will he will forget it after that. And this person will further tell it to the next person and he will forget it. By forget, I mean, that is all. He goes away from the picture.

I tell this fellow, this fellow, ask him to tell the next person and then walk away from the place. And the next person is asked to tell the next person and walk away from the place and so on. Do not you think using this method where a piece of rumor is supposed to spread to 100 people that is same as spreading it to just one person and asking him to spread to the other person just so that this way, the rest of them are handled to in the sense that it spreads to the last person here in the list. See this is recursion as common sensical as this may sound. This is indeed recursion.

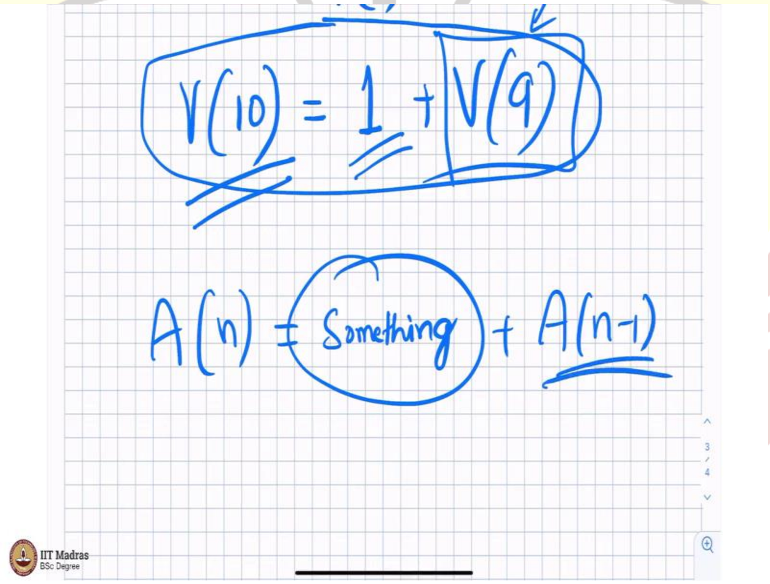
(Refer Slide Time: 03:43)



V(10) : Clean 1vessel
V(9)

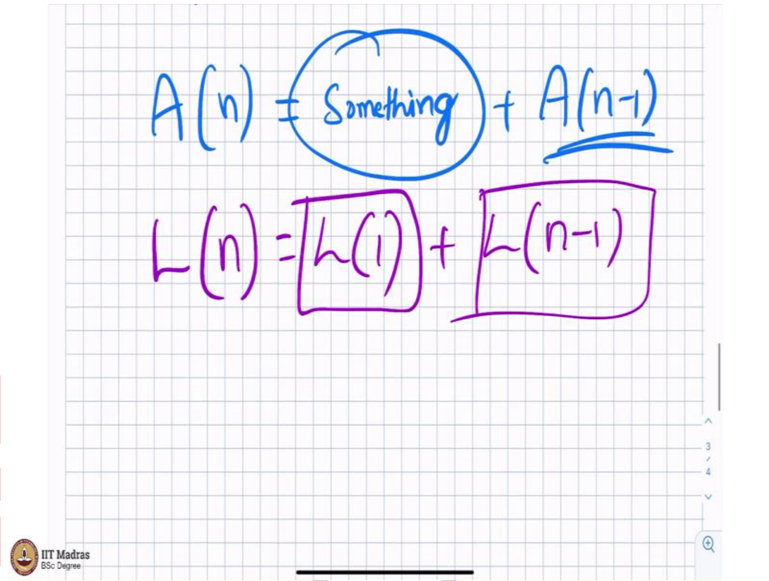
V(9) : Clean 1vessel
V(8)

V(8) : 2 vessels
V(6)



$$\boxed{V(10) = \underline{\underline{1}} + \boxed{V(9)}}$$

$$A(n) = \boxed{\text{Something}} + \underline{A(n-1)}$$



$$A(n) = \text{Something} + \underline{A(n-1)}$$

$$L(n) = \boxed{L(1)} + \boxed{L(n-1)}$$

So, let us go with another example just so that it is easy on our minds. So, assume you have thrown a dinner party to some of your B.Sc. classmates at the IIT Madras B.Sc. degree program. So, you have thrown a party and you have friends home. And obviously, when you throw a dinner party, your vessels require cleaning. So, you have some 10 vessels. 10 vessels are to be cleaned. And as always we procrastinate when it comes to doing tasks, especially household tasks. So, how would you go about cleaning these 10 vessels?

One way is to simply clean one vessel and request your family to help you with the other nine vessels. When you request let us say your brother to help you with the other nine vessels, your brother will sort of clean one vessel. And then ask, let us say, sister to clean the other eight vessels and the other eight vessels what is the sister does is she cleans two vessels and asks dad to help her with the other six.

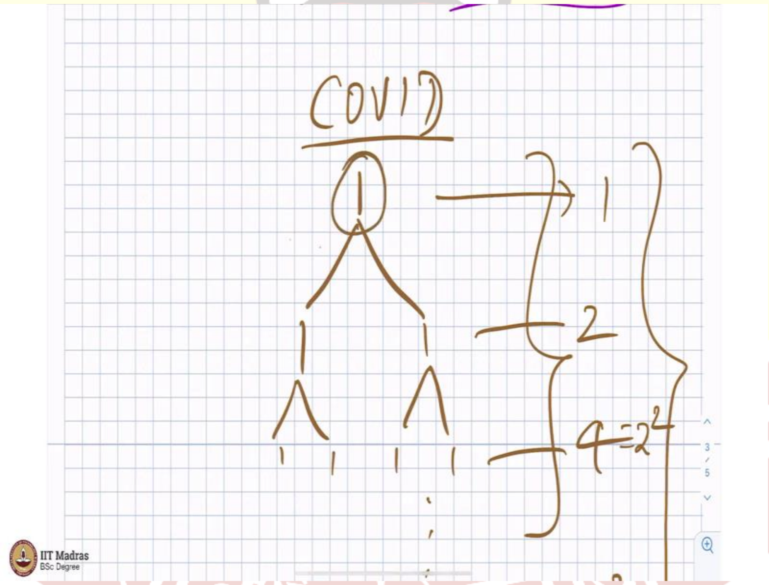
So, the idea is the problem of solving 10 vessels is same as the problem of solving one vessel, plus the problem of solving the nine vessels. It may not be an equation like this. I may be mathematically wrong to state it like this, but just analogously speaking, you see what this means. Washing 10 vessels is same as washing just one vessel and outsourcing the nine vessels to someone. This idea is very easy in your minds when it comes to programming.

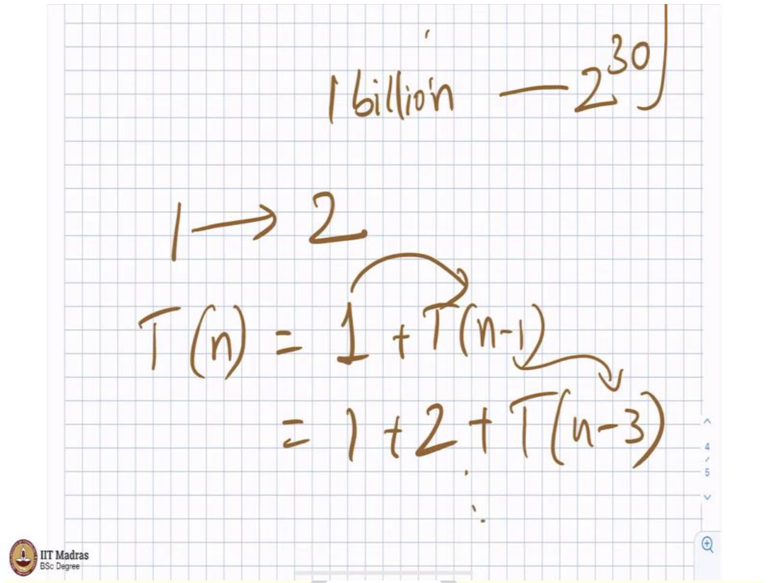
Many a times we may have to find the answer for n elements then it is enough if we can do something about one element and then outsource the other n minus 1 case to someone else. So, by that, I mean, in the queue, you see, we did this right. In the queue, I just spread the rumor to the first person and then asked him to spread it to the next person and then him to the next person and so on.

So, the problem of spreading the rumor to 100 people is same as the problem of spreading it to one person and then managing the rest 99 somehow. You see, you can always break the problem one thing at a time, as they say, live your life one day at a time. So, how do you live your life as stupid as this may sound, very relevant though in today's life. So, living for 10 days is the same as living for just today and then living for the other n minus 1 days.

You just worry about today and the rest you manage it day after. Therefore, the rest of your life how do you want to live, you manage later on. Maybe this is a bad suggestion, especially when it comes to today's world where saving is very important. We may have to think about day after, next year, next 10 years and things like that. But analogously speaking, even living for n days involves living for today to begin with so on and so forth.

(Refer Slide Time: 07:05)





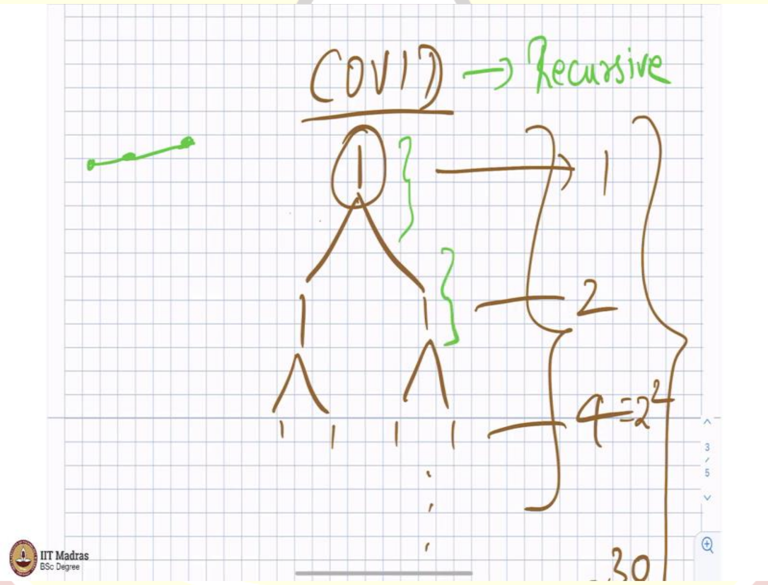
1 billion — 2^{30}

1 → 2

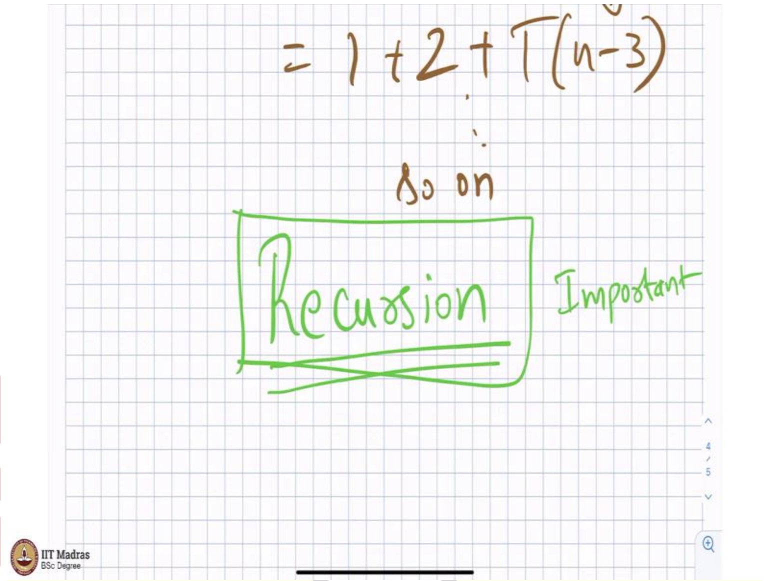
$T(n) = 1 + T(n-1)$

$= 1 + 2 + T(n-3)$

⋮



COVID → Recursive



$= 1 + 2 + T(n-3)$

So on

Recursion

Important

And I will give you one more example of recursion the case, the deadly case of COVID. This very coronavirus, it uses recursion to spread to the whole population. How, what, how does it do? Its target is to trouble n people. What it does is, simply trouble one or two people or let us say it troubles only one person. Let me just write it in a different way. So, it initially troubles one person and that one person at least infects two more people.

And these two more people, in fact, two more people and so on. So, at this stage, it is one, at this state it is two, this stage it is four. If you go on like this, it reaches roughly 2 to the 30, which is in the order of 1 billion. More than half of our population can get infected in just 30 steps. You see, 4 is 2 square, so on and so forth. Do not worry about the math here. The idea is coronavirus. What it does is it ensures that one person spreads to two people.

So, one person spreads to two people. The task of spreading it to n is handled the following way; first spread to one person and then think of spreading to the next n minus 1 people. How do you spread that? This one person will take two more people here. So, 1 plus 2 is done. And then n minus 1 becomes n minus 3, you see, so on and so forth. Do not worry if the math is confusing. In fact, this is not again, very mathematical.

I am just trying to give you a sort of a hindsight a bird's eye view of what happens in COVID infections. It is also recursive. COVID is also recursive. COVID uses a simple algorithm of I have spread the virus. Now, I have created a calamity inside the body of this one person. And now if he infects at least two people, I am done. One is not enough.

Why, because one person infecting only one person infecting only person that chain grows slowly, while this kind of a chain goes fast. I think this should be enough an introduction for recursion as and always programming involves practice. So, this very task of recursion is important for a simple reason that when you say when does programming gets complicated it is when there is a question on the recursion, because it is not easy in the mind.

You will see that in this week that we are going to do some non-trivial programming, not at all straightforward, but does not always whatever be it when you do it a couple of times three times, four times, five times, it becomes trivial. Everything is difficult before it becomes trivial and between difficulty and triviality there is practice. So, let us dive in to more details on recursion.