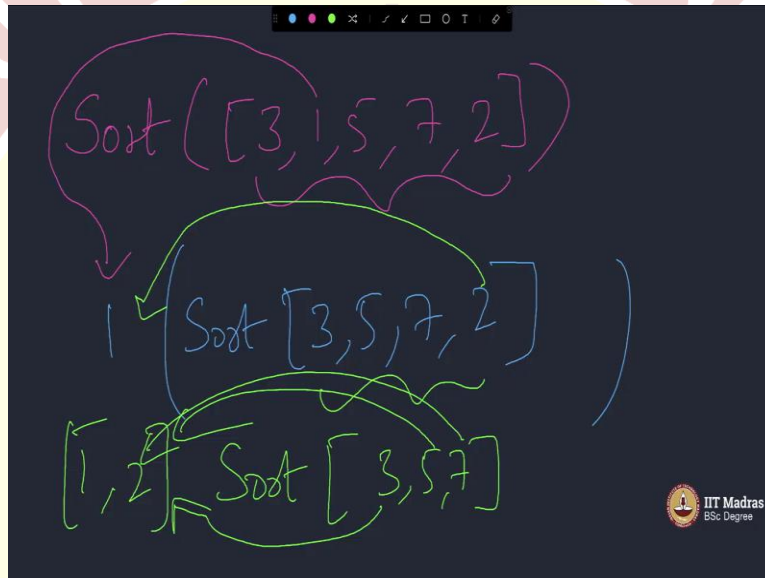




IIT Madras
ONLINE DEGREE

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
Sorting Recursively

(Refer Slide Time: 00:16)



Let us try to look at how one could sort using recursion. So, I need to sort let us say a list comprising of 3, 1, 5, 7, 2. How do I do that? One nice way to do this is to find out what is the minimum most element here, which is 1 and write that here. And after you remove this, you sort the remaining elements 3, 5, 7 and 2.

And then what happens, the minimum most element here, which is 2, again comes out, so 1 and 2 has come out, and you sort the rest, which is 3, 5, 7. You see, as you proceed like this, you will end up sorting the entire list. And after this the next step would be 3 will come here and then 5 will come and then 7 will come and bingo, you will have the list sorted. Very simple idea. I will not spend much time on explaining this.

(Refer Slide Time: 01:36)

```
1 def mini(L):
2     '''finds the minimum element in the list L'''
3     mini=L[0]
4     for x in L:
5         if (x<mini):
6             mini=x
7     return mini
8
9
10 def Sort(L):
11     '''recursively sort the list L'''
12     if (L==[]) or (len(L)==1):
13         return L
14     #if the list is empty, there is nothing to sort
15
16     m=mini(L)
17     #m now contains the minimum most element in L
18     L.remove(m)
19     #we remove that element from L
20     return [m]+Sort(L)
21     #we recursively sort the smaller list.
22
23 L=[5,6,59,19,2,1,3,10,11,121]
24 print(Sort(L))
~
"recursion.py" 24L, 531B written
[0] 0:Python*
```

In [5]: [1]+[5,7,6]
Out[5]: [1, 5, 7, 6]

In [6]: run recursion.py
[1, 2, 3, 5, 6, 19, 59]

In [7]: run recursion.py
[1, 2, 3, 5, 6, 10, 11, 19, 59, 121]

In [8]:

IIT Madras
BSc Degree

"Sudarshans-iMac.local" 01:44 21-Jun-21

So, let us go ahead and then try coding this. Probably it is not going to be easy for a first timer. But you now know how to use functions, you see. I mean, so you can slowly try to code it and try to figure out what errors you encounter and fix them on the fly. So, you observe that you need to sort a given list. How do you go about it?

First, you find the minimum element in the list. What is the minimum element? Minimum element will be, let me, I should basically create another function for it. Define mini, let me remove this, first I should find the minimum in the list given list L. How do you do that? I say mini equals L of 0. We have done this before. So, I will be a little fast here. Do not mind these error messages. So, error messages, I will tell you why it is coming.

If you put an empty list and try to say the program, it throws up an error, at least on the editor that I am using and the compiler that I am using right now. So, mini is L of 0. What I do is, for x in L if x is less than mini then my x becomes minimum, as simple as that. So, let us write some help commenting here that helps you to see the code after some time.

So, what do I say here, finds the minimum element in the list L, then I come here and I say x equals, I am sorry, what am I saying, mini equals x. These kind of mistakes we do. It is quite common. So, we should figure it out. That is all. And then what I do is and then I will return mini. So, please note, if in case x is less than mini, then mini becomes x, I go till the end of the list. Pretty straight forward code.

So, I will now come and what do I do? I should find out, I should sort the list L and for that, I should first find out what is the minimum element in L and then recursively sort it. So, how do I go about it? First, let me write a comment recursively sort the list L. How can recursively one can, how can one solve the, how can one sort a list recursively. Pretty simple.

All you do is you first extract the minimum most element here. How do you extract, m equals mini of L. So, as you can see, let me just show you here, one moment. So, you have this mini L here and I am going to find that mini L here as simple as that. I sign that to m. M will now contain mini and the function is written here. I am just calling it here.

And then the next step should be recursively sort. How do I go about doing that? In pretty simple as you can see. After this I will say I will return the sorted list, I will remove, first of all, m from it and then I will sort the remaining and attach m to it. So, this probably will take some time on your minds. So, let me try explaining what this is. This, as you know, is 5, 7, 6.

You see, this will concatenate these two lists. Put them together. So, that is what the plus here stands for. All that I am saying here is simply this. I am returning the minimum most element alongside the sorted list L with one element removed. That is it. This will do the trick now. You are seeing what is happening here, just stare at this again. So, if it helps, maybe I will write some comments so that it is easy for you people. The comment below is the comment about the above statement. Let us stick to the standard.

Maybe I have tried doing it the other way around. But from now onwards, we will try to see if we can stick to the standard of writing comment below the statement, m now contains the minimum most element in L. So, and then what does this do? We remove that element from L and then we recursively sort the smaller list. Why is it smaller, because one element is removed here as you can see. You have removed an element here and hence it is smaller by one element.

But then what if the list is empty? So, we need to take care of those things. What if the list is empty? Let me write that down. If the list is empty then I simply return the list. The list is sorted already correct. So, what does this mean? If the list is empty, there is nothing to sort. That is why you return it, return the list itself. What if the list contains only one element, even then there is nothing to sort. So, what you do is you say or the list contains only one element?

In that case, you simply return the list. So, probably, if the list is empty or if the list has only one element, there is nothing to sort, you simply return the list. Otherwise, you go ahead find the minimum and then remove that minimum element and then return recursively the element appended, you see what happens when you do this, when you put a plus, you put, you include that minimum most element and then sort the rest recursively this is done. I am suppose you have understood functions. This should not be difficult then.

So, this should work fine. Let us see if it does not work, we will fix it. So, what do we do? Let me just print something here. Let me call L as 5, 6, 5, 9, 1, 9, 2, comma 1, comma 3. Let we now print sort of L. Let us see, when I run recursion is the file name, I see that this is getting sorted, 1, 2, 3, 5, 6, this gets, this is getting sorted. So, we can consider making the list bigger and so on. Still you see that it is getting sorted. No element is missing.

I just counted that the list contains 10 elements. So is the sorted list. So, so far, so good. You understand how you can recursively sort a list. So, at this time, at this instance, you can skip the rest of the video and go ahead with the next video. But then for people who are finding it difficult to follow, let me go through it just one more time.

So, what am I doing? Firstly, I write down a small function which finds the minimum of a given list L. Now, I am going to use that here. What am I going to do? I simply extract, pluck out the minimum most element in L here, remove that from L, because I want L to be one element smaller, because one chore is done, one task is done, minimum most element comes out, keep that minimum most element in the beginning and sort the rest of the list, that is it.

Absolutely simple, straightforward code and you will get the answer here. So, that is pretty much it. Let us go ahead and try writing something that is not this simple. So, what could that be? Let us try to see how we can search for an element in a given list. Now, what is so difficult about it? Let us see the next video. It is not very easy, at the same time it is not also very difficult. So, I am going to introduce you people to a brand new searching technique in the next video.