INDIAN INSTITUTE OF TECHNOLOGY MADRAS

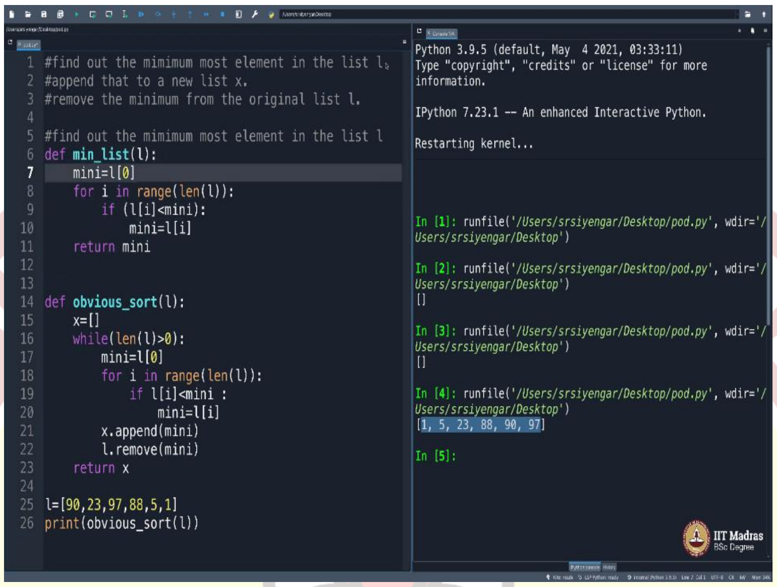
सिद्धिर्भवति कर्मजा

IIT Madras

ONLINE DEGREE

Programming in Python
Professor. Sudarshan Iyengar
Department of Computer Science and Engineering
Indian Institute of Technology, Ropar
Sorting Using Functions

(Refer Slide Time: 00:16)



```
#find out the mimimum most element in the list l,
#append that to a new list x.
#remove the minimum from the original list l.

#find out the mimimum most element in the list l
def min_list(l):
    mini=l[0]
    for i in range(len(l)):
        if (l[i]<mini):
            mini=l[i]
    return mini


def obvious_sort(l):
    x=[]
    while(len(l)>0):
        mini=l[0]
        for i in range(len(l)):
            if l[i]<mini :
                mini=l[i]
        x.append(mini)
        l.remove(mini)
    return x

l=[90,23,97,88,5,1]
print(obvious_sort(l))
```

```
Python 3.9.5 (default, May  4 2021, 03:33:11)
Type "copyright", "credits" or "license" for more
information.

IPython 7.23.1 -- An enhanced Interactive Python.

Restarting kernel...



In [1]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')

In [2]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')
[]

In [3]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')
[]

In [4]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')
[1, 5, 23, 88, 90, 97]

In [5]:
```

Remember, the sorting program that we did the last week, I am going to repeat the same right now, but I am going to use functions and show you the power of functional approach to programming. If you remember, we sorted a list of numbers based on a technique called the obvious sort.

If you know, if you if you remember what that was, it was find out the minimum most element in the list, append that to a new list, and then remove that, remove the minimum, from the original list. Append that to a new list, let us say new list x, find out the minimum most element in the given list let us say l, and remove the minimum from the original list l. This is what we did.

I will try to do this right now. On purpose, I will stumble a little, I will do it slightly slowly. And then try to tell you the power of functions and try to sort of demonstrate how both time and effort gets reduced if you use functions. Let us go ahead and try to code the obvious sort algorithm.

So, I will say obvious sort, I will take l as the input. So, what should I do, look at the first line, find out the minimum most element in the list. Fine. How do I do that? How do I find out the minimum most element in the list? You see, I am thinking, I am sort of clueless. As I told you earlier I am going to take a very, very slow approach to solve this. So, assume you are new to programming and you are thinking aloud like this with me.

So, how do you sort a list l, you first need to find the minimum most element in l. How do I do that? I will say for i in range length of l. So, if minimum that you have declared already, minimum, which is the first element in the list, if whatever was the minimum is if l of i is greater than, I am sorry, less than you see how much I am struggling. l of i is less than minimum, then your minimum will be your l of i, that is what we did.

But then at the end of this, you will just get the minimum most element, as you come out of the for loop. So, what you should do is you must append that to a new list x, let me declare that you list here, x equals this new list. And then what so I am clueless slightly right now, what should I do. I think, I removed the element mini from the list l. This was the program that we wrote in the previous week videos.

Let me think what happens, x is an array empty array minimum will be declared as the first element in the list and then I will use a for loop to find out if there are elements less than the minimum. In case I find one, then I will call that as the minimum and at the end of this for loop, you would have found out the minimum most element in the list and assign that to mini. And then I will append that to x, and I will remove that element from l, this entire thing should actually be in a loop.

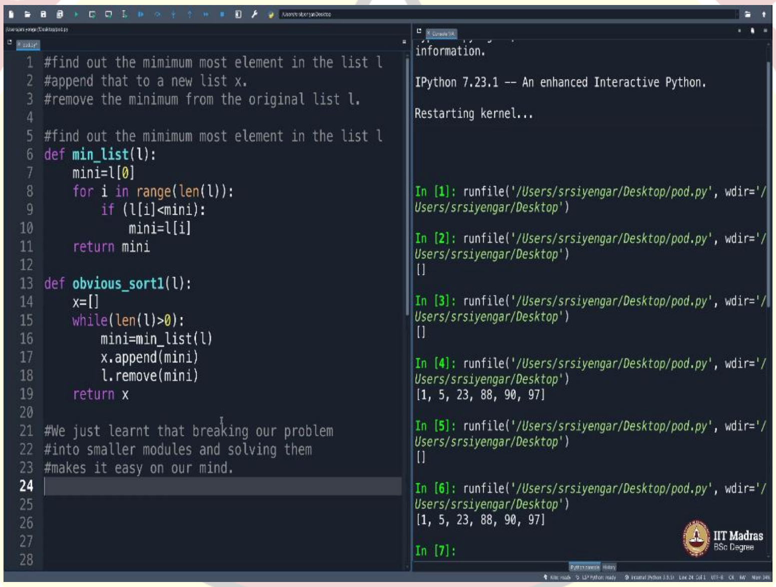
It should be in what loop? While loop? Why? I keep doing this until the list becomes empty. You are removing the entries from the list. Length of l is not 0. By that I mean, it should be 1 or above then I keep doing this. And as you are removing elements from l, it will go to become empty finally, and then finally you will come out of the loop, and then I will say return l. hopefully this works. Let us see.

Let me say l equals 90, 23, 97, 88, 5, 1, so I need to sort this. Let me say the new let us say print obvious sort of l let us see what this does. It is not printing anything for some reason, let us see y, okay, return l, you see the mistake I did. I should have returned x here because x is the new list. You probably were wondering as I was coding. Some of you on seeing this mistake, you are wondering what is he doing, and that was seen here, my stupidity.

So, now it sorts the list. 15, 23, 88, 90, 97. And you are now wondering, what on earth did you do just now? The same old program that we did the last time I am, just a minute, I am using functions to do the same thing. What is the big deal in it? There is a big deal in it, just wait and watch. In fact, I wrote the entire thing into one function, but then look at this, I have find out the minimum most elements in the list l. Why cannot I write a function for this? So, what do I do?

I will just do, I will just write a function exclusively for this. What is that? For this, find out the minimum most element in the list l so let me call that as define min, minimum element in the list l, this should return a minimum element after finding it. So, how do you find it, you say minimum equals l of the first element and for i in range length of l, you say if length l of i is less than mini, then your minimum will be l of i. So, the moment you do this this function simply gives you the minimum most element in the list.

(Refer Slide Time: 06:45)



```
#find out the mimimum most element in the list l
#append that to a new list x.
#remove the minimum from the original list l.

#find out the mimimum most element in the list l
def min_list(l):
    mini=l[0]
    for i in range(len(l)):
        if (l[i]<mini):
            mini=l[i]
    return mini

def obvious_sort1(l):
    x=[]
    while(len(l)>0):
        mini=min_list(l)
        x.append(mini)
        l.remove(mini)
    return x

#We just learnt that breaking our problem
#into smaller modules and solving them
#makes it easy on our mind.
```

```
information.

IPython 7.23.1 -- An enhanced Interactive Python.

Restarting kernel...



In [1]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/Users/srsiyengar/Desktop')

In [2]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/Users/srsiyengar/Desktop')
[]

In [3]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/Users/srsiyengar/Desktop')
[]

In [4]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/Users/srsiyengar/Desktop')
[1, 5, 23, 88, 90, 97]

In [5]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/Users/srsiyengar/Desktop')
[]

In [6]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/Users/srsiyengar/Desktop')
[1, 5, 23, 88, 90, 97]

In [7]:
```

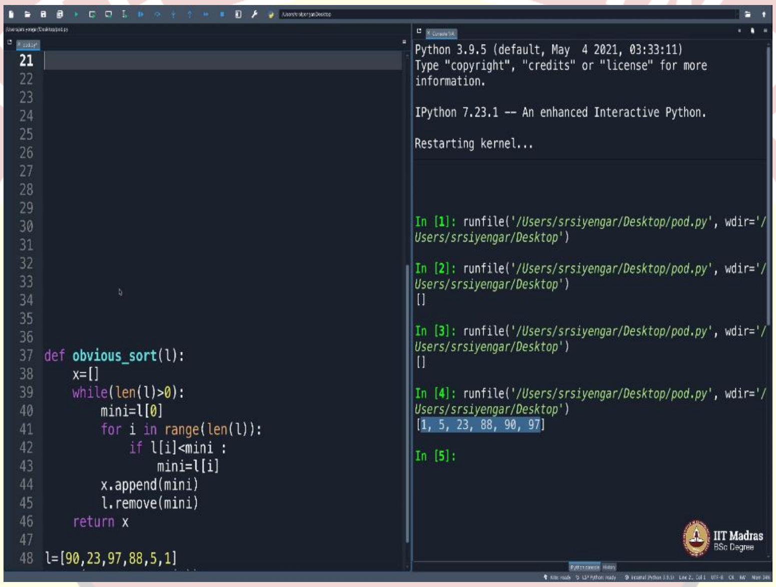
And now, I will go ahead and write a sorting technique. So, what I will do is I will make this disappear for you, let us not see it. And now, let me write down first part is over, append that to a new list, remove the minimum from the original list. So, now let me say, this is a new sorting technique.

So, I will say obvious sort 1, a different name as a function. I input l, and I want it sorted. So, what do I do, first, I find out the minimum most element in the list l, which is mini, which is equal to min list, you see, this is the climax here, minimum list l simply puts the minimum most value of l into mini. You see, as opposed to what I did here. I had to write a program for it. Now, that is not required.

You simply put the minimum most element in of the list l into mini. And then what I will do is, I will append that mini to a new list x. Of course, this all should be declared here, x is so much. X and I will remove the element mini from and l that is all over. This entire thing should come in a loop, you see, correct, think, that is precisely what I was doing. initialize x to empty, and then find out minimum, append that to a new list x and remove that element from the list l.

Keep doing this while your list is not empty. Same program, you see how organized we were in our thinking, correct. And then finally, I say return l. Over, I feel this was easier for me to do than the previous sorting technique.

(Refer Slide Time: 08:32)



```
def obvious_sort(l):
    x=[]
    while(len(l)>0):
        mini=l[0]
        for i in range(len(l)):
            if l[i]<mini :
                mini=l[i]
        x.append(mini)
        l.remove(mini)
    return x

l=[90,23,97,88,5,1]
```

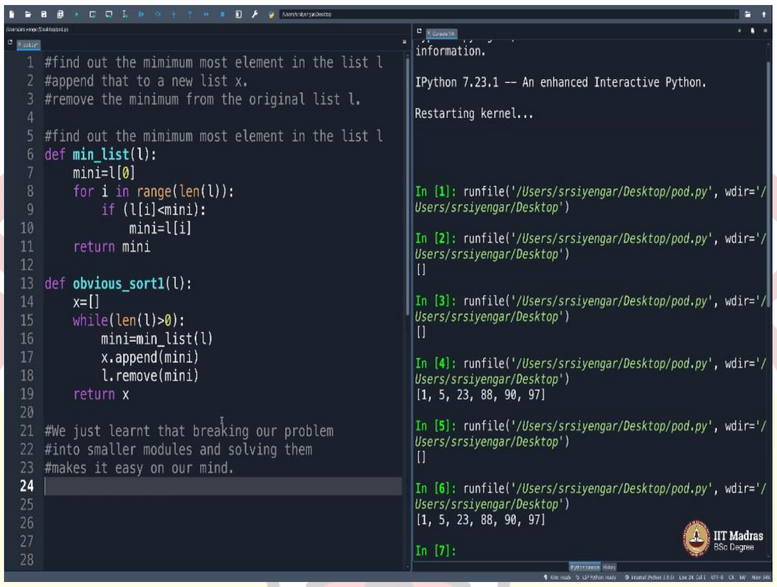
```
Python 3.9.5 (default, May  4 2021, 03:33:11)
Type "copyright", "credits" or "license" for more
information.

IPython 7.23.1 -- An enhanced Interactive Python.

Restarting kernel...



In [1]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')

In [2]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')
[]

In [3]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')
[]

In [4]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')
[1, 5, 23, 88, 90, 97]

In [5]:
```

This was a little garbled here. I mean, it was a little heavy on the mind, you see, but this was not that heavy on the mind. Correct, so let us see whether it works fine. What I will do is, if I say obvious sort this gets executed, I will try to say obvious sort 1, and ii will see which means it will come here, execute this function. Inside this function, there is another function here, which it will go and then call. So, this is like you seek the help of your friend. So, this part, you seek the help of this fellow that is how it works.

And here, you are calling obvious sort 1. Let us see if it works or not. Again, I did the same mistake, you probably kind of thinking how many times will Sudarshan keep doing return x instead of return x I typed return l. So yes, I see the answer here. Perfect. So, what is the moral of the story?

The moral of the story is a big program can be made into bits and pieces and you can solve them slowly. Is it not true in your life too, where you have a big task to do and you keep procrastinating, and a good tip that people give you is break that into a smaller into smaller chunks and then try to solve these chunks independently, individually, that way it becomes easy on your mind. You will not procrastinate. You will feel organized and you will get the job done. So, we reached here with understanding a very important principle in programming, which is functional approach to programming, also called the modular approach.

(Refer Slide Time: 10:15)



```python
#find out the mimimum most element in the list l
#append that to a new list x.
#remove the minimum from the original list l.

#find out the mimimum most element in the list l
def min_list(l):
    mini=l[0]
    for i in range(len(l)):
        if (l[i]<mini):
            mini=l[i]
    return mini

def obvious_sort1(l):
    x=[]
    while(len(l)>0):
        mini=min_list(l)
        x.append(mini)
        l.remove(mini)
    return x

#We just learnt that breaking our problem
#into smaller modules and solving them
#makes it easy on our mind.
```

```
information.

IPython 7.23.1 -- An enhanced Interactive Python.

Restarting kernel...



In [1]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')

In [2]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')
[]

In [3]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')
[]

In [4]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')
[1, 5, 23, 88, 90, 97]

In [5]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')
[]

In [6]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')
[1, 5, 23, 88, 90, 97]

In [7]:
```

You break your problem into small modules, let me just sort of write that down. We just learnt that breaking our problem into smaller modules, and solving them makes it easy on our mind. So, if you were to ask me what is the most important part of our discussion so far, I would say it is the ability to break a program onto smaller chunks, and that is precisely what you have learned so far. Please go ahead and practice, practice and practice, try to write all the code that you have seen so far, using functions.

Now we will go ahead and try to use functions to see if the very process of matrix multiplication which was one big task for us very heavy, an ordeal and very tough on our mind can be simplified that using a functional approach to multiply two matrices. We will see that in the next video.