

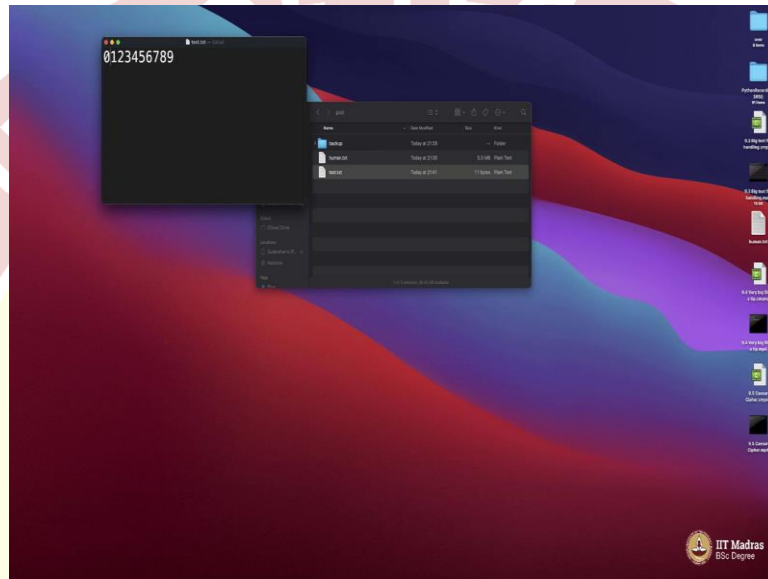


IIT Madras

ONLINE DEGREE

Programming in Python
Professor. Sudarshan Iyengar
Department of Computer Science and Engineering
Indian Institute of Technology, Ropar
Mr. Omkar Joshi
Course Instructor
Indian Institute of Technology, Madras
Online Degree Programme
File Handling, Genetic Sequences

(Refer Slide Time: 0:16)



```
In [5]: ls
backup/ human.txt test.txt

In [6]: f=open('test.txt','r')

In [7]: s=f.read(2)

In [8]: print(s)
01

In [9]: s=f.read(2)

In [10]: print(s)
23

In [11]: s=f.read(2)

In [12]: print(s)
45

In [13]: f.seek(4)
Out[13]: 4

In [14]: s=f.read(2)
t
In [15]: print(s)
45

In [16]:
```

```
In [10]: print(s)
23

In [11]: s=f.read(2)

In [12]: print(s)
45

In [13]: f.seek(4)
Out[13]: 4

In [14]: s=f.read(2)

In [15]: print(s)
45

In [16]: f.seek(2)
Out[16]: 2

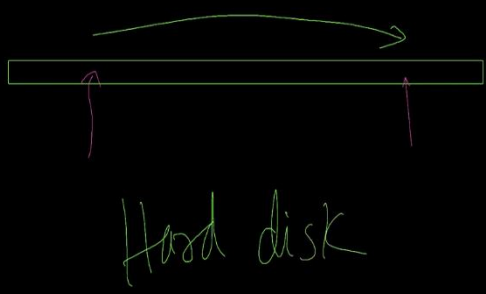
In [17]: s=f.read(1)

In [18]: print(s)
2


In [19]: f.seek(7)
Out[19]: 7

In [20]: f.read(1)
Out[20]: '7'

In [21]: ]
```



Hard disk



So, I have these two files here, test dot txt, let me see what is in text dot txt, it has only these 1,2,3,4,5,6,7,8,9 and 0. I need to teach you people something, so I am using this file. It is simply, it has this 10 elements. Maybe we can put 0 in the beginning so that it's easy for you people and I save it. So, test dot txt, let me open this file, open test dot txt, in readable format.

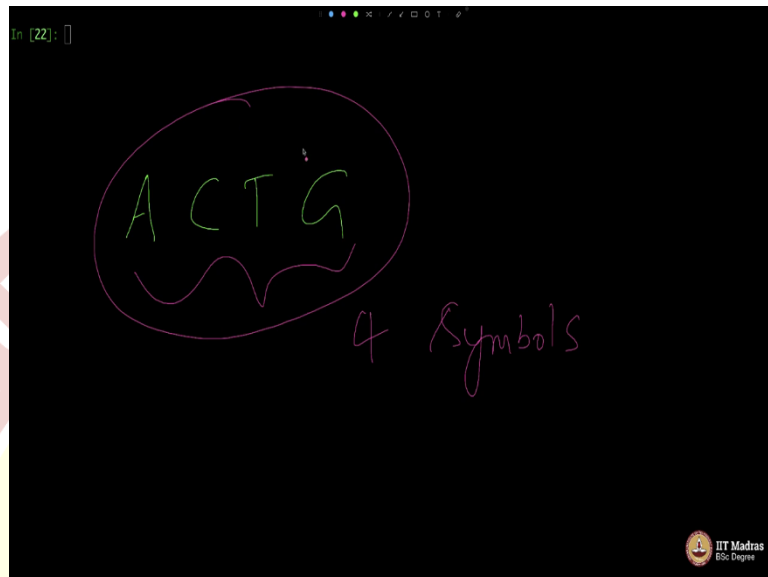
Whenever I say, let us say S equals f read 2, it will read two characters. Again, if I say S equals f read 2, it will read the next two characters 0,1,2,3. Again, if I do this, it will read the next two characters. So, what if I say f dot seek 4. What does it do? Let us say S equals f read 2 characters and print S. It goes and sits in this position and reads the next two letters, in the fourth position. The fourth position is 0,1,2,3 after that, the next two letters is 4 and 5.

So, let me go and sit in. So basically, I am rewinding back the file. Go into the second position, and then I am reading the one character let us say and printing that, it is 2. When I go to the seventh character f seek, and then print or simply say f read of 1 there it is, sum. So, f seek basically helps you go forward or go backward. You can directly go to that one place in the file. It is not very easy, it does not go magically, there it goes, sort of linearly.

Now what do I mean by that by that? By that I mean, if the file is, let us say a big one it is a big file if you use f seek, it directly goes to the place of your choice. If you say, go ahead, it will come here, but it will go linearly. By that, what do I mean by that? By that I mean, from here to here, it will manually move and then come here. It does not go suddenly here. So, it can be costly sometimes f seek can be costly because it depends on how your hard disk stores the information.

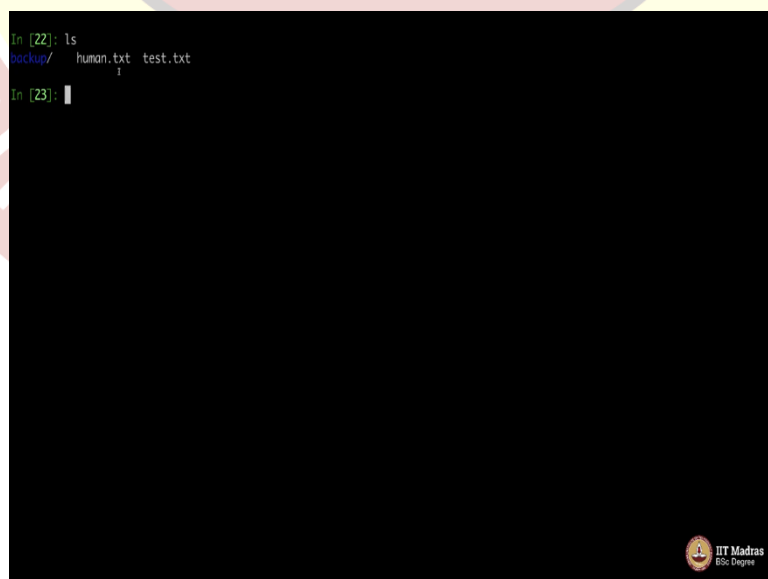
Anyway, technicalities aside, the point is use `f seek` judiciously. Expect it to take a long time, if you input a big number here. Keep that in your mind. I am not going to go into the details of `f seek`, but you may want to use it when time comes.

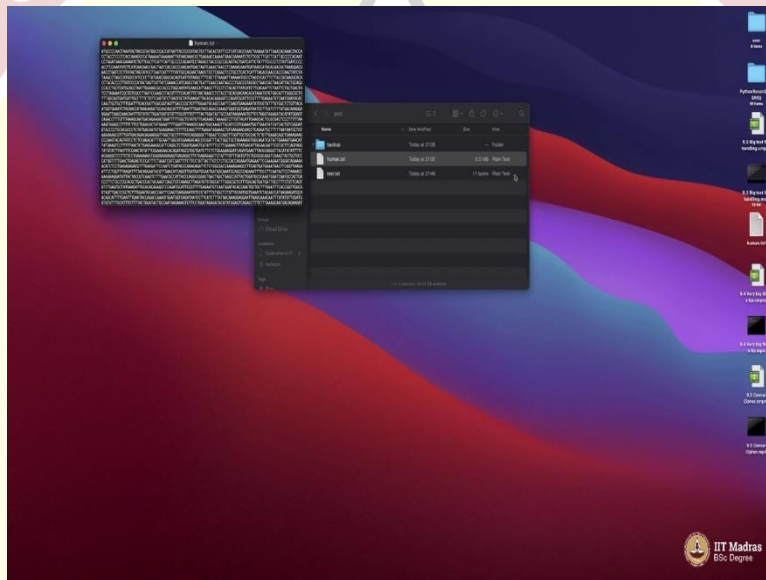
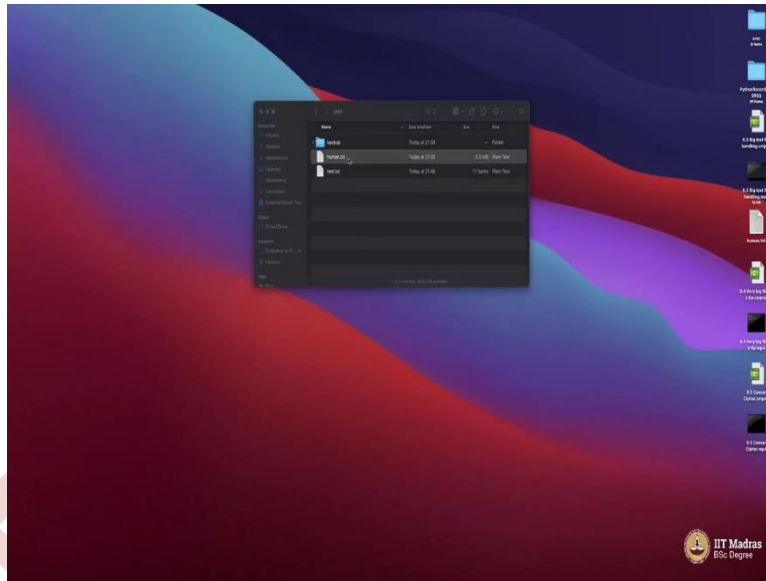
(Refer Slide Time: 3:28)



Fine, let us now go ahead and do something very interesting and something biological. So, your entire genetic sequence of a human being comprises of A C T and G these 4 symbols only. If you get your genetic sequence, it will have A C T G in some form in some sequence, a long sequence you will have. Let us see one such sequence now.

(Refer Slide Time: 4:00)



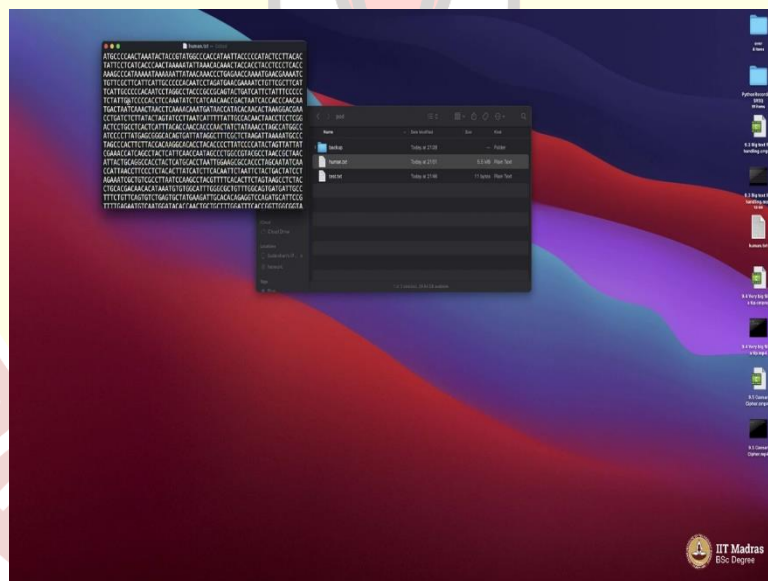


I have it with me, I have downloaded this from an online data set. I have the data set with me, let me open that, what is that, human dot txt is the file, f equals let me open the file and show it to people. Let me close this human dot txt if you click, you will get this big file. Not very big as you can see only 5.5 Mb that is some human genome sequence for educational reasons I am using it.

I may not be very accurate with what I explained about the biological aspects of what is coming next. So, please pardon me in case I am not very accurate, but it is a very, it is just a motivational example. I am zooming in it is taking some time. Let it take its own sweet time. Let us wait for a couple of seconds. So, zooming in is taking time.

No problem as you can see, the entire sequence is made up of ACTG. You see. This looks like some sci-fi movie. If you have seen the movie Matrix, this is how it appears, although, green in color, now we have white in color here. Again, I am zooming in so that you can see the screen you only have sequence of A C T G appearing randomly.

(Refer Slide Time: 5:35)



In the sequence, if a particular sequence is present then you may you may declare that the, I do not know why it got closed. It is here, so in particular sequence comes then it may mean that this this body is predisposed for diabetes, if some sequence comes it means it is blood pressure. Some sequence comes, it may appears it means, it is predisposed for COVID-related infections.

(Refer Slide Time: 6:17)

So, what I will do is, I will open this big genome sequence in readable format. Good, and what I will do is very simple, I will call it `sequence` is equal to `f.read`. `F.read` simply takes and puts it into a string `a seq`, `print seq`, it will print. It is a big string.

(Refer Slide Time: 6:46)

```
GTGTGCACATTATCAG'
In [26]: seq[0]
Out[26]: 'A'

In [27]: seq[1]
Out[27]: 'T'

In [28]: seq[2]
Out[28]: 'G'

In [29]: diab='GTATGAC'

In [30]: diab in seq
Out[30]: True

In [31]: bp='TAGAACCTGGATA'

In [32]: bp in seq
Out[32]: False

In [33]: 'GTA' in seq
Out[33]: True

In [34]: 'GTAC' in seq
Out[34]: True

In [35]: 'AAAA' in seq
Out[35]: True

In [36]: 'AAAA' in seq

In [35]: 'AAAA' in seq
Out[35]: True

In [36]: 'AAAA' in seq
Out[36]: True

In [37]: 'AAAAA' in seq
Out[37]: True

In [38]: 'AAAAAA' in seq
Out[38]: True

In [39]: 'AAAAAAA' in seq
Out[39]: True

In [40]: 'AAAAAAA' in seq
Out[40]: True

In [41]: 'AAAAAAA' in seq
Out[41]: True

In [42]: 'AAAAAAA' in seq
Out[42]: True

In [43]: 'AAAAAAA' in seq
Out[43]: False

In [44]: 'AAAAAAA' in seq
Out[44]: True

In [45]: 'AAAAAAA' in seq
```

Seq of 0 now will be the first letter, seq of 1 will be the second letter, seq of 2 will be the second, third letter and so on. Let us say that diabetic gene is will have the sequence GTADGAC I just cooked that up it will not be this small, it will be very big. Actually, I am not a biologist. But still, the idea is fairly simple.

If there is a sub, there is a string like this visible in the genome sequence, it means that you are predisposed for a particular disease. By predisposition I mean, you may get it. There are high chances that you will get it. So, it will not be this small of course, it will be very big but for educational reasons, I am going to use it very small.

If I diab in seq, true. This particular sequence is there in a seq. Let us say bp assume bp was TAGAACCTGGATA. So, bp in seq false, it is not there. So, see in general if you have a

Let me keep adding and see, how bigger contiguous A is present in this sequence still A there somewhere there is the sequence with 1,2,3,4,5,6,7,8,9 is very good 10 is also there, 11 is also there. Is it 12, 1,2,3,4,5,6,7,8,9,1,0,11,12, 12A is not there.

[illegible][illegible]

11 A is there 12 As are not there, which means in the sequence, there is some way that you can see 4 Ts are here. All I am saying is 12 As is what was that? I forgot. 12 As are not there but 11 As are there somewhere. Very interesting. So, you can find, if there is some, in fact, it

will be a very big sub-sequence in a human being. Every human being will have a genetic sequence like this. Again, I am putting it in very, very simple terms. In fact, it is not as simple as this. The genomic sequence is different. It is made up of many such sequences. But to make it easy on your mind I am just trying to make, have a very simplistic model.

A sub string like this, if it is present then you say this particular disease, probably this gene can have and so on. But did you observe something? How did my program go? I said `f equals open`, and then I say `S equals simply really, correct`. What was that? `Seq equals, sequence equals` I simply said something like `f equals read`.

If the file is very small this command gets executed in no time as you saw, and `seq` becomes a string. But if this file is a big one, if this file is a big one, it takes some a few days for it to enter into the memory. In that case, what you should do is you should go letter by letter here. You should go letter by letter by using `f dot read 1, f dot read`, followed by this stands for the number of bytes that you are going to see. So many bytes at a time.

I think you get the spirit right now, maybe you can write a piece of code that takes a very big, few gigabytes, big sequence, and then tries to find a subsequent there. There is a very popular method called very ingenious method. In fact, Knuth-Morris-Pratt K-N-U-T-H, Morris, M-O-R-R-I-S, Pratt, P-R-A-T Knuth-Morris-Pratt algorithm to find a substring in a given big string. It is a very ingenious algorithm.

It is a very ingenious algorithm you can take a look if you are interested. Of course, this is not part of your syllabus that we will be discussing. But if you are very curious how this is done, take a look at this. This is really, really a five-star algorithm, really something that is my favorite. Take a look at this. And that is pretty much a file handling this week. All the best, go ahead and learn file handling well. This this will come in use to you for a lifetime.

In fact, whenever you handle big data or huge datasets from online, you may require file handling extensively. With this, I end the file handling session. And now we will now take up Pandas and we will teach you a bit of Pandas and after that the week ends. Thank you all very much.