Physics through Computation Thinking

Dr. Audítya Sharma & Dr. Ambar Jain

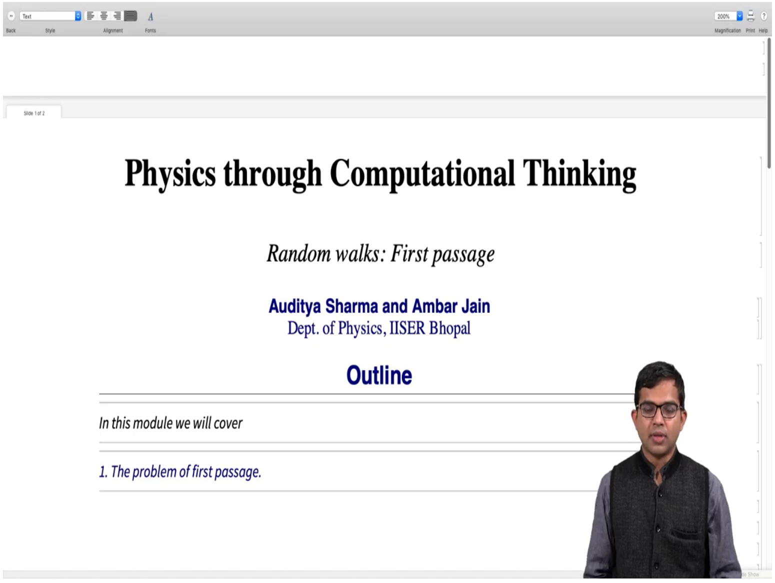
Department of Physics

Indian Institute of Science Education and Research, Bhopal

Lecture 50

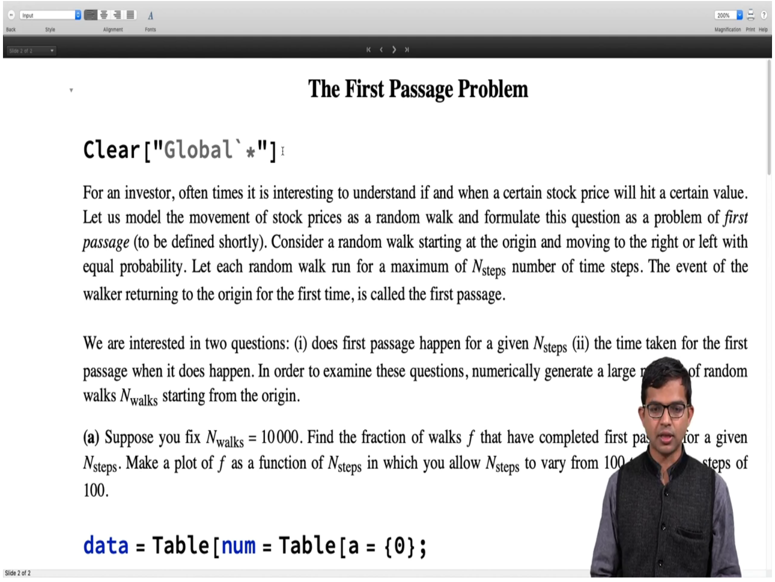
Random Walks 7

(Refer Slide Time: 0:27)



Hello everyone, so this is an offshoot of the discussion about Random Walks. I want to cover one application which I think is very nice and very neat and with a little bit of mathematical skills one can already see some very interesting results. So, if you want to work it out analytically you will have to work much harder. So, I want to show you this about the so called first passage processes in the problem of the random walk.

(Refer Slide Time: 0:58)



So, as always, I will start by clearing Mathematica of any prior knowledge. So, as you can see that I have already run Mathematica, so it is showing some weird number. If it had started afresh it would have been just 1, but it is showing 223 at the moment but that does not matter. So, I clear all the variable information it has.

So, as you can see on the screen, so this problem pertains to, for example, that of a stock investor. Think of a stock investor who has put some money in some piece of stock at some point and then they find that after some time the value of the stock has dropped. So, they are like why did we put this money? And at least even if it comes back to where I started, I am willing to sell it so that I do not have any loss.

If I am unable to make any profit then let me atleast not make a loss. So, the stock, the investor is hoping and praying that the stock would come back to where it started. So, this question of whether if you model the stock prices as a random walk, if the random walk will again pass through the origin or any particular point that you wish to choose is a problem of first passage.

So, you can think of the 1D random walk who starts off at the origin and does some complicated path and then you ask whether they come back to the origin at all and if they do, how much time will it take? These are the two questions that the stock investor would be interested in; one is whether if it is a random walk whether, unbiased random walk, whether it will come back at all to a certain point and if it does, then what is the typical time that you would have to wait before it happens? So, these are the two questions and so we will model this as a pure unbiased random walk.

But in reality it is really not the correct model for the stock market. It is way more complicated than the simple 1D unbiased random walk like here, but let us understand this question for the 1D random walk problem. We have already discussed how if you take N steps on already you have covered a distance of order $\sqrt{N}$ and we looked at many ways of seeing this result and how this is a ubiquitous result and how this is connected in general to the problem of diffusive motion and so on.

So, now this question is about does first passage happen at all, that is the first question and the second is the time taken for first passage. So, let us numerically just generate a large number of walks and keep track of it and work out the solution. So, I have broken down this problem into a bunch of things to do. So, if you want you can pause the video at various points like here now; at this problem A.

You can pause the video all you can get a glimpse of my solution but it is not really going to help you. You can pause the video and try to work this problem out for yourself. So, suppose you say that you want to generate 10000 random walks, unbiased random walks, we have already seen how to do this.

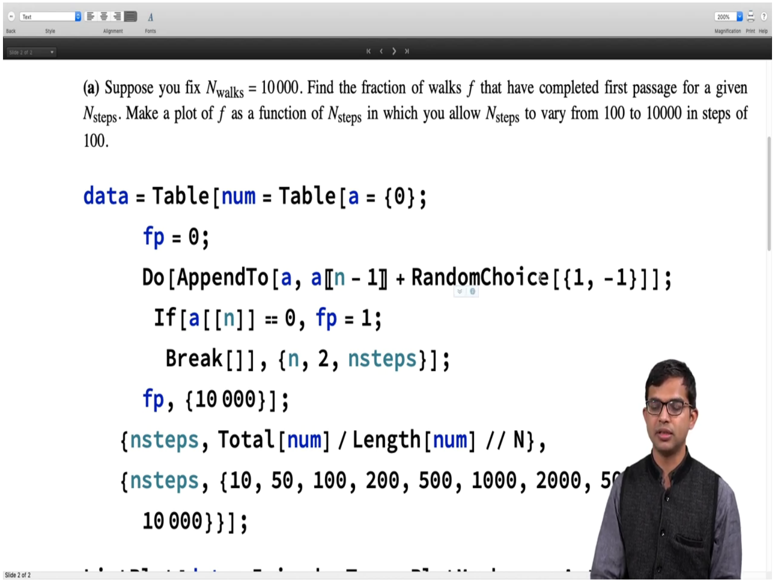
Now, find the fraction of these walks that have completed first passage for a given N steps. So, let us say that you keep running this random walk for N steps and you ask whether it has returned to the origin for the first time within those N steps. If it does then you count those walks and you divide by the total number of random walks.

And just make a plot of; you know, out of 10000 random walks how many of them have returned to the origin within N steps and then you increase N steps and then you keep on varying it and see as a function of N steps what is the fraction of random walks which will return. So, this will in fact give us an estimate for what would happen if for a typical random walk.

We are imagining a scenario where you have like really large number of random walks. That is one way of thinking about what happens in a distribution. Okay, so here is my solution. If you have tried enough and if you feel that you already have a solution you can look at my solution or if you feel that you have tried enough and you have given up, you can also look at my solution.

But I prefer that you really work hard before one way or the other. You must make sure that you have given your best attempt before you look at my solution.

(Refer Slide Time: 5:16)



So here is my; not necessarily the best solution but it is a solution and I know that it works because I have checked it. So, what I do is I generate a table and inside this I have another table. So, this table will generate for me a random walk. You have seen how to generate a random walk; we have done it in the past.

So, I start with an array with just 0 in it. It is just one element array and then I am introducing this quantity which I am calling fp; $fp = 0$ and then do append to this array of size. To this array you append a_{n-1}; the last element of this array plus one random choice; either plus 1 or minus 1. Okay, so here is my solution to the problem.

So, what I do is I have a table of tables. Now, this inside table is something that you have something sort of you have already seen because I am generating a random walk here and I am calling it num, for reasons will be evident to you in some time. So, there is a table which contains a equal to just one element which is just 0 and you have seen how this works.

And I am introducing this thing called *fp*, *fp* for first passage equal to 0 and then I have a random walk, append to this a, a_{n-1} plus random choice of either plus 1 or minus 1, you go to right or left. And now what I do is if at some n, n has gone to 0, this n will start only from 2. So at $n = 1$, of course it is already at 0 and you can clearly see that it will never be at 0 for an odd number of steps.

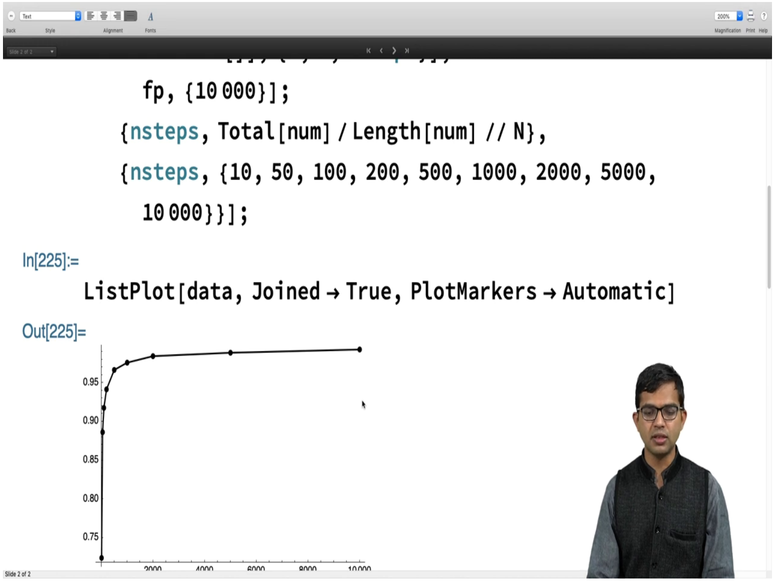
In the first step it has to be at plus 1 or minus 1, second step it can be at the origin. Third again it cannot be at the origin, fourth it could be and so on, only even steps are possible but that is some detail. So, here I am allowing n to go from 2 to *N* steps and if $a_n = 0$, so there is an If command within Mathematica, you can look up the syntax if you are interested.

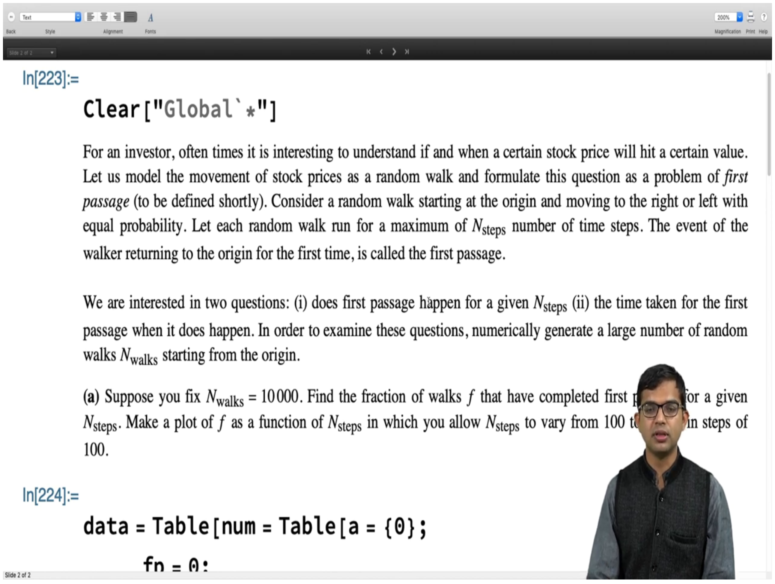
$a_n = 0$, then I will put $fp = 1$, first passage has happened basically and then I break this. I do not want to keep doing this do loop. And then within this *N* steps if *fp* happens *fp* becomes 1, otherwise it will remain 0. And then I just note down *fp* and then I do this 10000 times; that is why 10000 such random walks.

Then I will note the number of *N* steps. Then I will do the total of num divided by length of num which is actually nothing but the mean of num. I could have directly done mean of num you will get the same answer, it does not matter. And then I will allow *N* steps to take all these values; 10, 50, 100, 200, 500, 1000 and so on all the way up to 10000. You can play some other set of numbers if you want. So, let me go ahead and run this.

It is still running. It will take a moment and then we can look at the data. So, what does, you know, the I have this num. Now, you know why I use num because I have to create find its mean. And then I create a whole table of this. Like exactly like what the problem statement describe from. Okay, so the job is done. Now, I am of course going to go ahead plot this, using list plot and there you go.

(Refer Slide Time: 8:42)

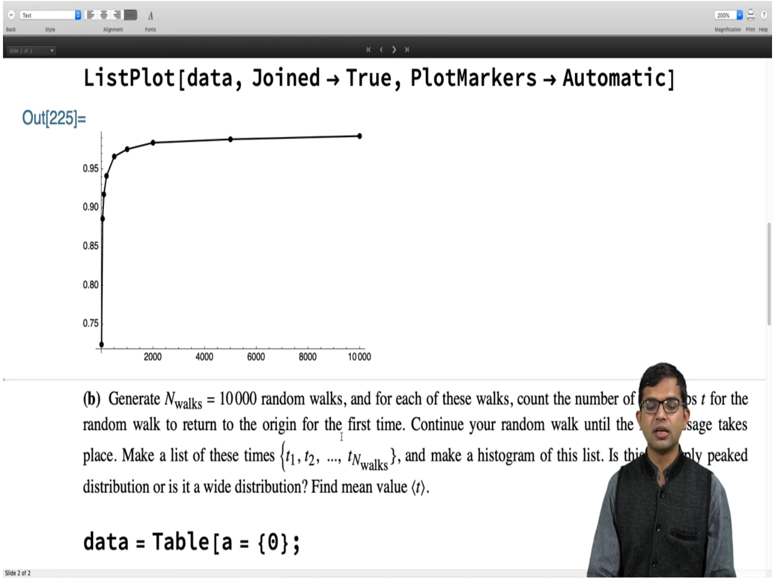




So, I have this fraction as a function of N steps. So, you see that this fraction is going to become as close to 1 as you want. You just have to keep on increasing N steps, right. In other words, if you are willing to wait for long enough, it seems like this probability that there is first passage is actually 1. In other words, with unit probability, there is going to be first passage no matter what. So, so the stock investor would just need to wait forever. For you know if he is willing to wait for long enough, then there is going to be first passage. Then, I was a bit hasty in saying something like forever.

So, let us actually ask ourselves, what is the typical amount of time that he has to wait before actually first passage happens? First statement is, the unbiased random walk is guarantee to undergo first passage.

(Refer Slide Time: 9:44)

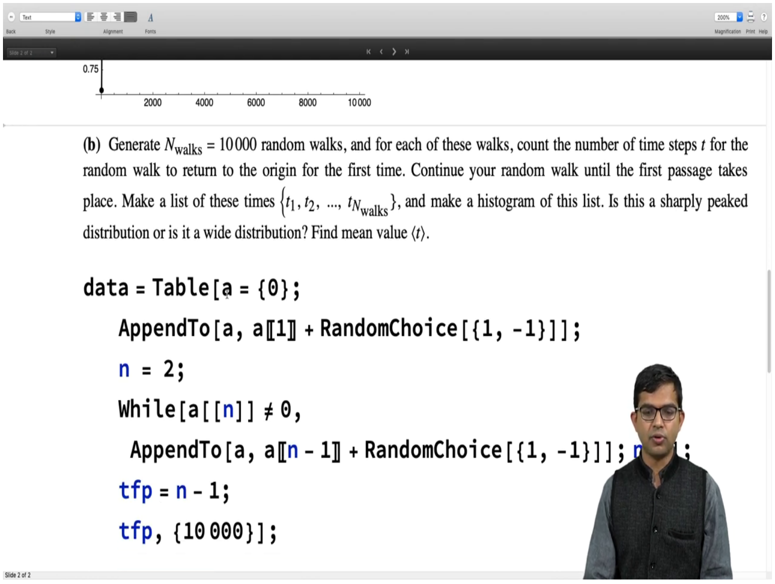


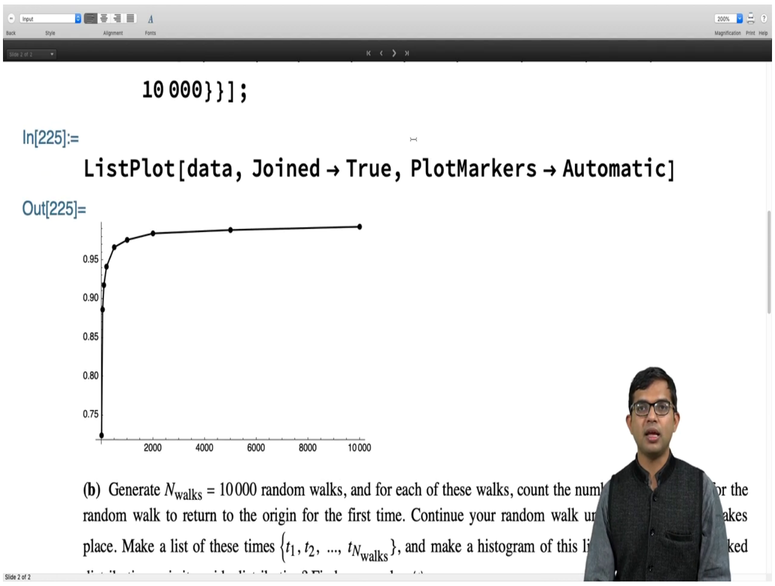
Now, the second statement is, what is the typical time? So, that is part B. so, here again once again you can pause the video and work this out for yourself. So, in other words, what you do is, generate 10000 random walks and for each of these random walks you just count the number of times step t for the random walk to return to the origin for the first time.

Continue your random walks until first passage takes place and just simply make a list of all these numbers and see what are these numbers like? Are these numbers which seem to have a nice average or is it like what is the distribution of this that is what you want to play with and find out. This is a sharply peak distribution. It is a wide distribution. What happens to the mean and so on? So, this analysis I want you to do. So, it is not very difficult you can use the code in the previous section and with some small tweeks, you can work this out.

So, I urge you strongly to pause the video, write your own piece of code, check this out and then only look at my solution. So, my solution is the following.

(Refer Slide Time: 10:44)





So, I will you know like before I have this table starting with 0 and then AppendTo I will do once, only once. Because I want to use this while command. So, $While[a_n \neq 0]$, I will keep on continuing this. Otherwise, the moment a_n becomes 0, I will just automatically stop. I do not want to use break like I did before, I can directly use a while loop. While $a_n \neq 0$ just append this and continue the random walk and then $n++$.

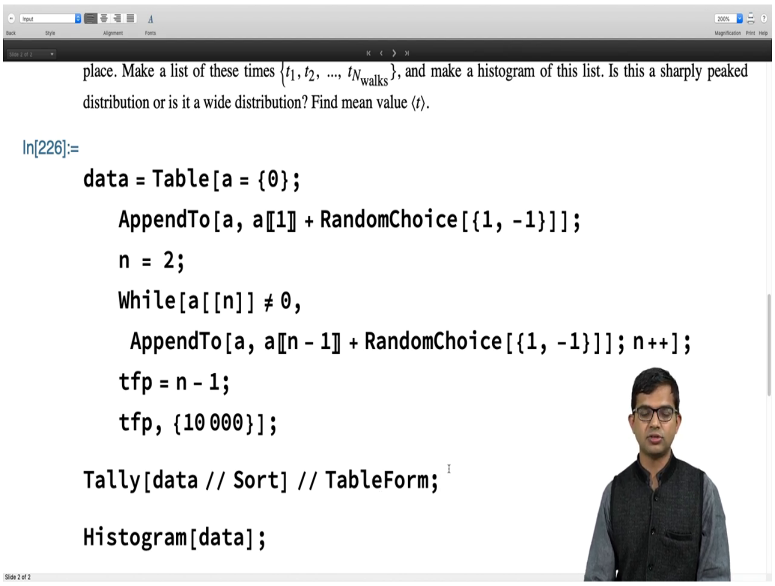
So, in order to start with $n = 2$, I generated my random walk from, starting from 0. And then one step I have already generated here. And then I put $n = 2$. And then while $a_n \neq 2$, clearly when $n = 2$, a_n is guarantee not to be 0. It is only from $n = 3$ onwards, it is possible.

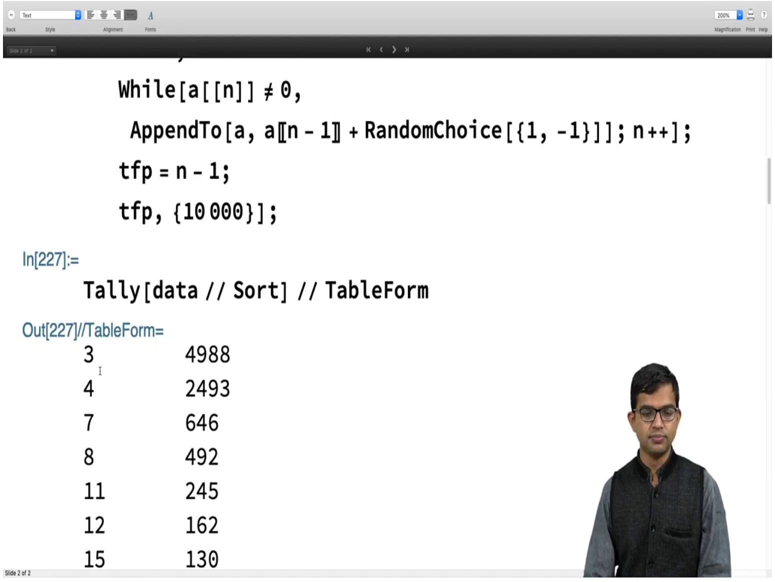
So, *tfp* will, will refer, will be $n - 1$. The first step. So, n and *tfp* are indeed related by this $n - 1$. If at the N step to you reach the time taken is $n - 1$. So, it is just the matter of some book keeping. Now if *tfp* happen I will keep a record of this first passage time and then I will do this 10,000 times.

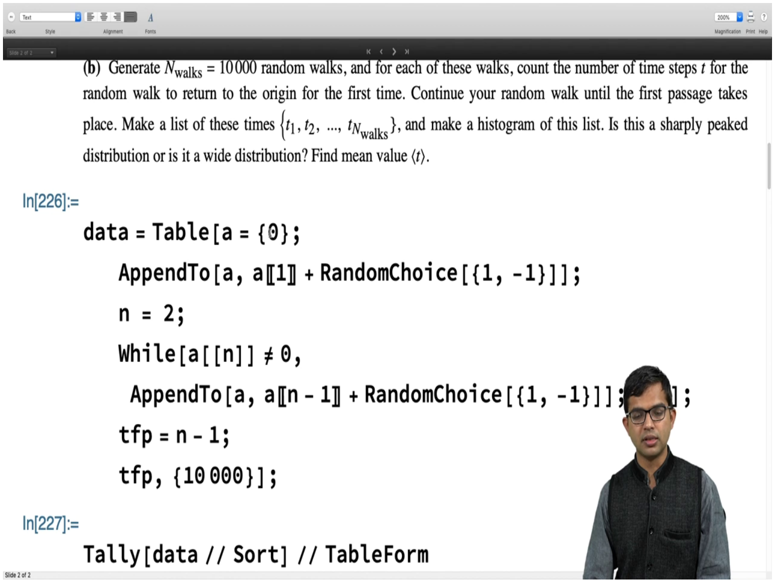
Right, I have, I am, generated 10,000 random box and I am keeping track of, so, my part A gives me the confidence that my simulation will actually stop. I mean in general, you do not want to put in a while loop like this like what if this condition is never met and then you may run into a loop which will never close.

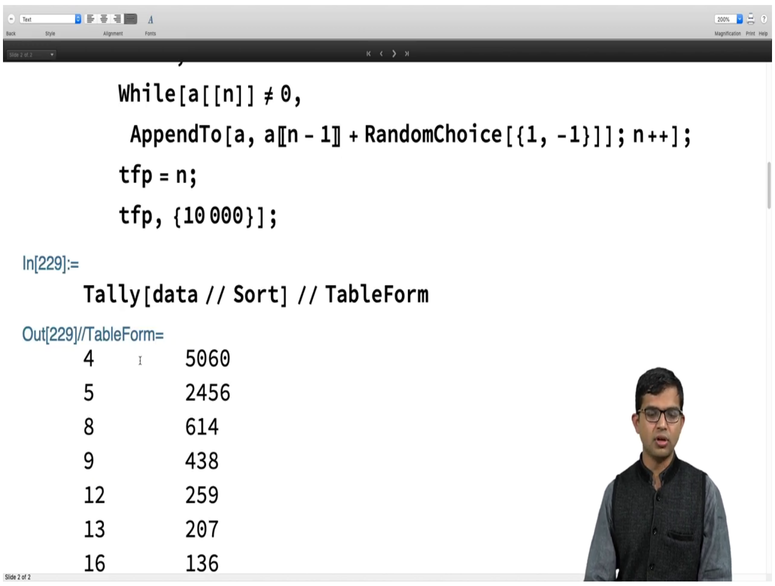
But my experiments with part A actually give me the confidence that if I can do this and I am telling you that it is possible. So, it is eventually you are going to get this *tfp*. So, let me go ahead and run this, and show you that indeed all of these random walks actually come back if you keep running the solution, simulation for long enough, it will come back. And so, let me write down, note down all this data. So, let me so, it is already done.

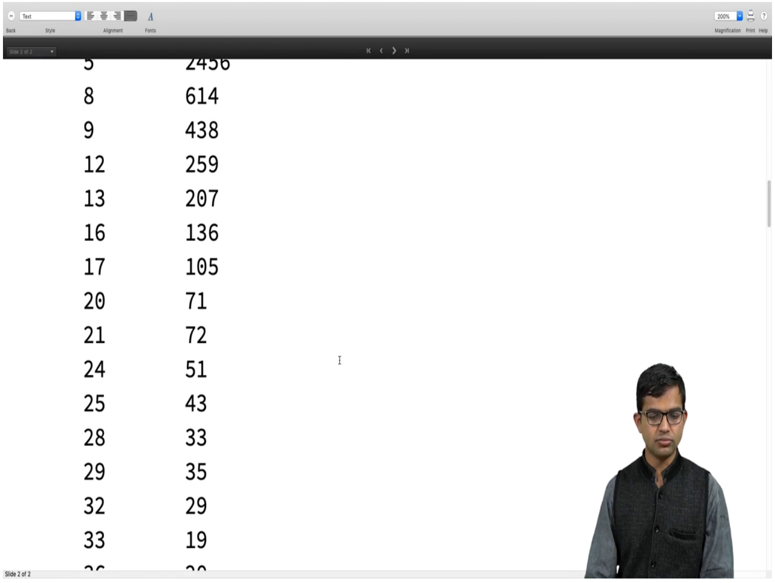
(Refer Slide Time: 13:02)




While[a[[n]] ≠ 0,
AppendTo[a, a[[n - 1]] + RandomChoice[{1, -1}]]; n++];
tfp = n - 1;
tfp, {10 000}];
In[227]:=
Tally[data // Sort] // TableForm
Out[227]//TableForm=
3 4988
4 2493
7 646
8 492
11 245
12 162
15 130


(b) Generate N_{walks} = 10 000 random walks, and for each of these walks, count the number of time steps t for the random walk to return to the origin for the first time. Continue your random walk until the first passage takes place. Make a list of these times $\{t_1, t_2, ..., t_{N_{walks}}\}$, and make a histogram of this list. Is this a sharply peaked distribution or is it a wide distribution? Find mean value $\langle t \rangle$.
In[226]:=
data = Table[a = {0};
AppendTo[a, a[[1]] + RandomChoice[{1, -1}]];
n = 2;
While[a[[n]] ≠ 0,
AppendTo[a, a[[n - 1]] + RandomChoice[{1, -1}]];];
tfp = n - 1;
tfp, {10 000}];
In[227]:=
Tally[data // Sort] // TableForm


While[a[[n]] ≠ 0,
AppendTo[a, a[[n - 1]] + RandomChoice[{1, -1}]]; n++];
tfp = n;
tfp, {10 000}];
In[229]:=
Tally[data // Sort] // TableForm
Out[229]//TableForm=
4 5060
5 2456
8 614
9 438
12 259
13 207
16 136



So, for 10,000 so, now if I were to so, I can use this thing called tally of data, sort the data and then tally and put it in table form. So, what it does is, it tells me that you know $tfp = 3$ happened 4988 times and *tfp* equal to. So, if *tfp* is the time of first passage. So, in the step 1 it is at 0 so, after one step it is going to be at the first. So, probably I should just take $tfp = n$ itself. So, then it going to be even.

So, it is something that it is not really going to substantially change the nature of the problem. You can play with this. So, let us make this equal to n and then we will see that it gives me a table form. So, it gives me all even numbers. So, then you have you know all this various possibilities are there.

(Refer Slide Time: 14:06)

In[228]:=

```
data = Table[a = {0};
   AppendTo[a, a[[1]] + RandomChoice[{1, -1}]];
   n = 2;
   While[a[[n]] ≠ 0,
    AppendTo[a, a[[n - 1]] + RandomChoice[{1, -1}]]; n++];
   tfp = n;
   tfp, {10 000}];
```

In[229]:=

```
Tally[data // Sort] // TableForm
```

Out[229]//TableForm=

4	5060
5	2456

```
   tfp = n;
   tfp, {10 000}];
```

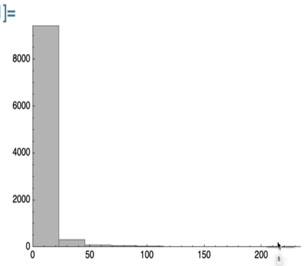
In[230]:=

```
Tally[data // Sort] // TableForm;
```

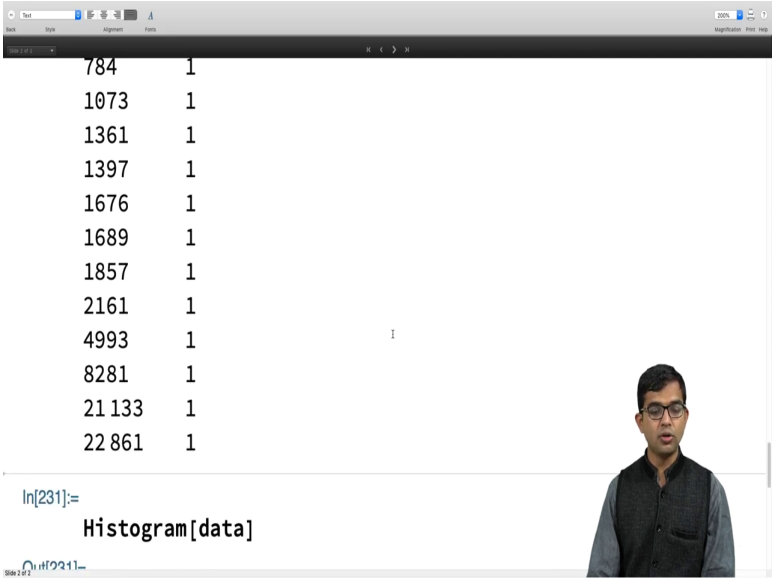
In[231]:=

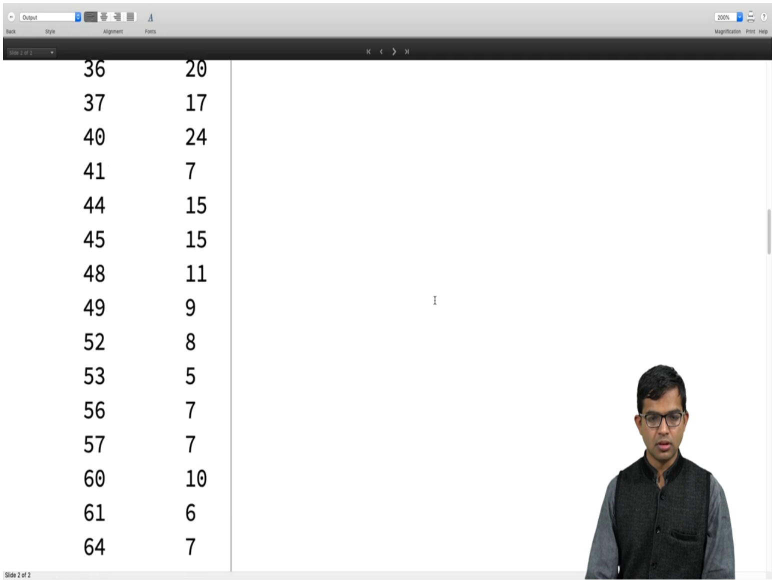
```
Histogram[data]
```

Out[231]=



44	15
45	15
48	11
49	9
52	8
53	5
56	7
57	7
60	10
61	6
64	7
65	6
68	5
69	8
72	7

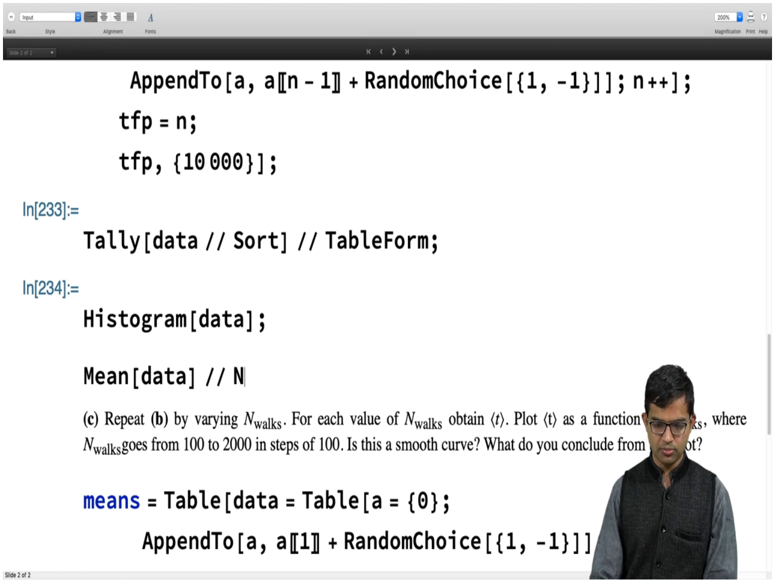




And then if I, I will hide this information and then if I do a histogram of this then you see that, actually you get a, you get representation even for very, very large numbers. And that is the point so, that is also evident in this, in this way of looking at the data. You see that some of these random walks actually run very, very, very long.

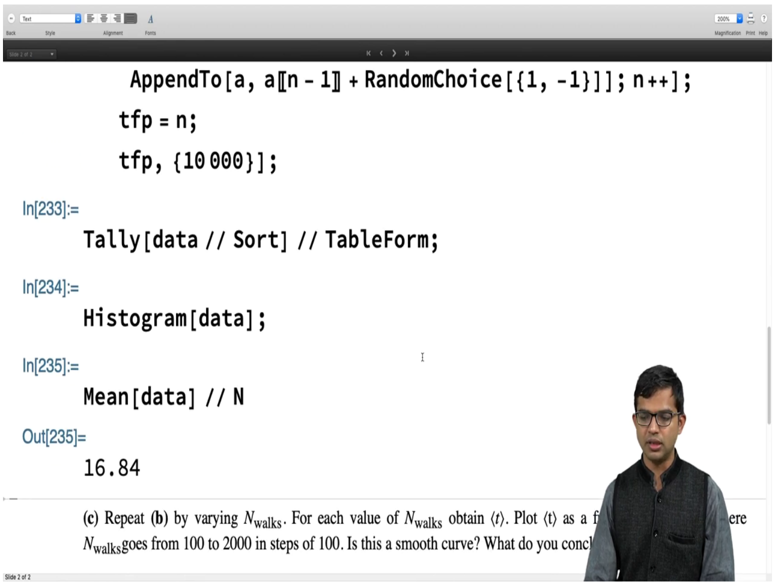
So, you see that there are 22,000 it has run for 22,000 steps before it has come back. Okay, I notice that some of these numbers are even and others are odd. And which seems to be an issue and so this is perhaps some small logical fallacy in this and so that would be homework problem to figure this out, but in essence this is indeed correct. So, you can work this out and then you have a histogram here.

(Refer Slide Time: 15:24)



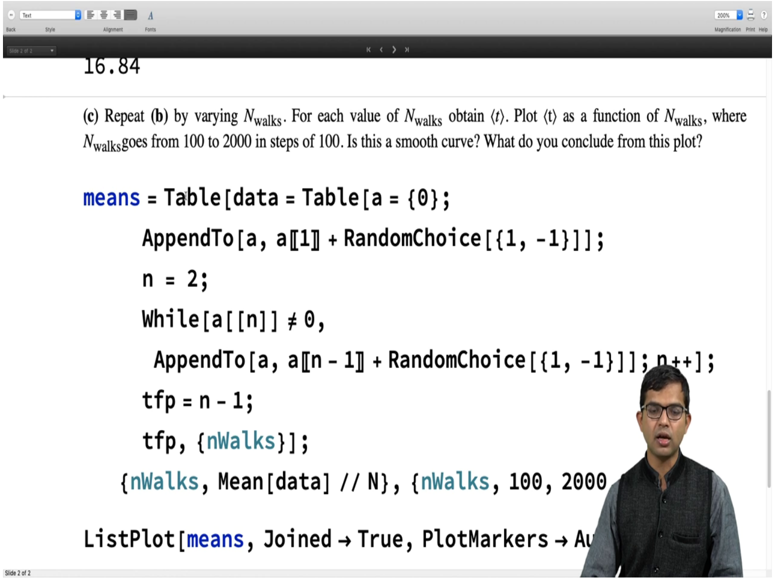
And so mean if I were to compute this mean, I get 16.8. So, third part of this is to repeat this whole thing and find out whether this set of means will itself converge or not?

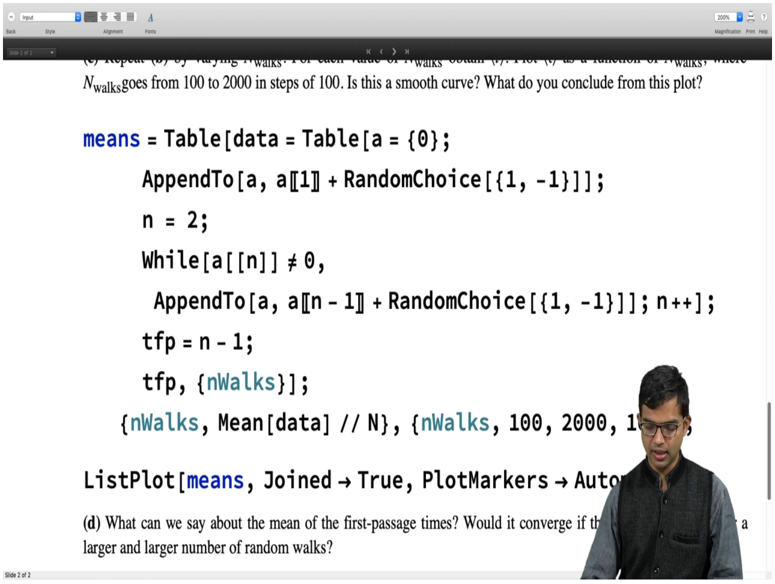
(Refer Slide Time: 15:45)

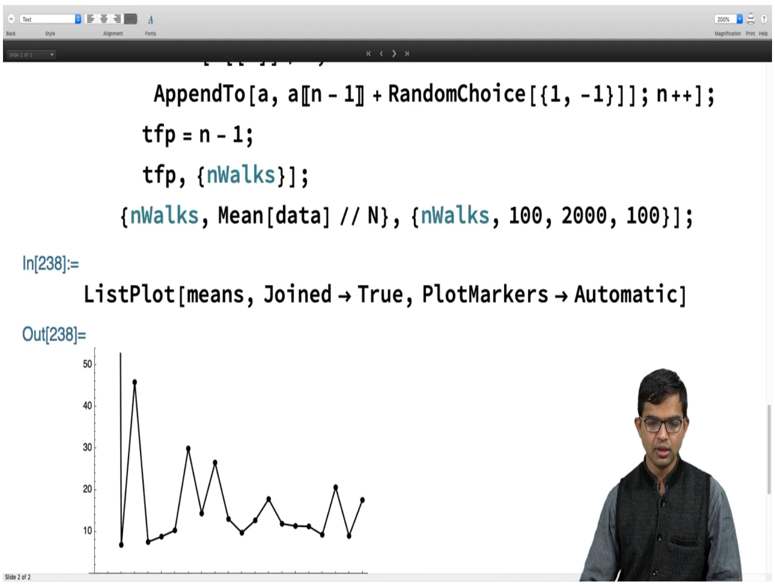


And in order to do this I have a part C associated with this and so, so part C what does it do? It says, you repeat part B by just varying the number of walks, *nWalks* itself and then you, you ask whether there is some convergence of this average value. So, that is what this part C does.

(Refer Slide Time: 16:08)


16.84
(c) Repeat (b) by varying Nwalks. For each value of Nwalks obtain ⟨t⟩. Plot ⟨t⟩ as a function of Nwalks, where Nwalks goes from 100 to 2000 in steps of 100. Is this a smooth curve? What do you conclude from this plot?
means = Table[data = Table[a = {0};
AppendTo[a, a⟦1⟧ + RandomChoice[{1, -1}]];
n = 2;
While[a[[n]] ≠ 0,
AppendTo[a, a⟦n - 1⟧ + RandomChoice[{1, -1}]]; n++];
tfp = n - 1;
tfp, {nWalks}];
{nWalks, Mean[data] // N}, {nWalks, 100, 2000
ListPlot[means, Joined → True, PlotMarkers → Au


Nwalks goes from 100 to 2000 in steps of 100. Is this a smooth curve? What do you conclude from this plot?
means = Table[data = Table[a = {0};
AppendTo[a, a⟦1⟧ + RandomChoice[{1, -1}]];
n = 2;
While[a[[n]] ≠ 0,
AppendTo[a, a⟦n - 1⟧ + RandomChoice[{1, -1}]]; n++];
tfp = n - 1;
tfp, {nWalks}];
{nWalks, Mean[data] // N}, {nWalks, 100, 2000, 1
ListPlot[means, Joined → True, PlotMarkers → Auto
(d) What can we say about the mean of the first-passage times? Would it converge if th
larger and larger number of random walks?

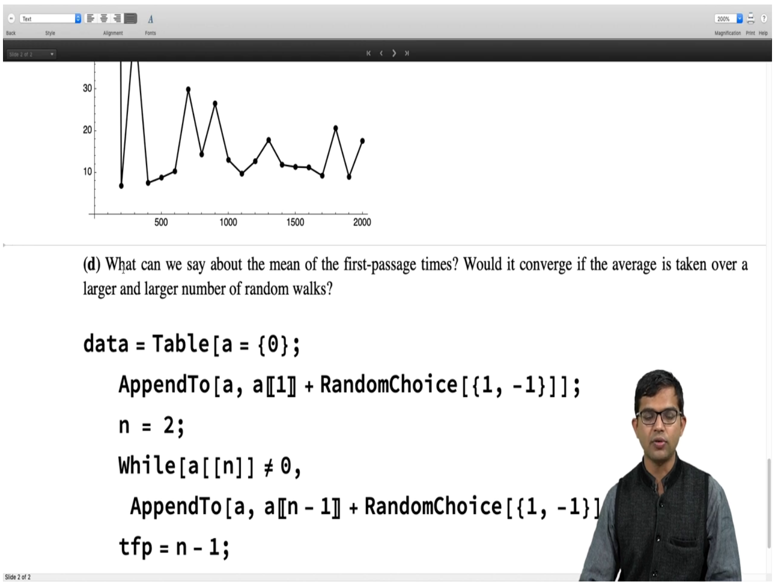

AppendTo[a, a⟦n - 1⟧ + RandomChoice[{1, -1}]]; n++];
tfp = n - 1;
tfp, {nWalks}];
{nWalks, Mean[data] // N}, {nWalks, 100, 2000, 100}];
In[238]:=
ListPlot[means, Joined → True, PlotMarkers → Automatic]
Out[238]=
50
40
30
20
10

So, my solution what I do is I will take all these means and then simply keep on increasing this *nWalks*. So, this is some codes that you can look at. So, here I have $tfp = n - 1$. And then I have a list plot. So, you see that, so this problem is going to take some time to run. So, first I have to run this. It is going to be, so, I hit shift enter it takes some movement. So, now when it is running I will explain what is going on.

So, I have a table of this. So, this is a, the inner table indeed what we have already seen. And then once we get this information for *tfp* for *nWalks* itself, I am taking it to be a random variable. And that I will allow it to go from 100 to 2000 in steps of 100. And now I am ready to plot it.

So, now you see that if I were to average over 100 or 200 or 300 and so on all the way up to 2000 of this. It is actually a wildly fluctuating number this mean of this quantity. It is not really there is no simple order to it.

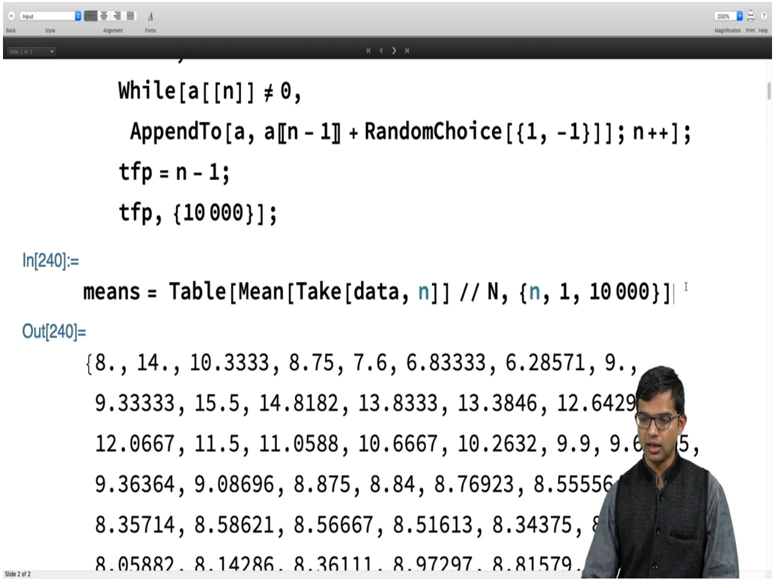
(Refer Slide Time: 17:13)

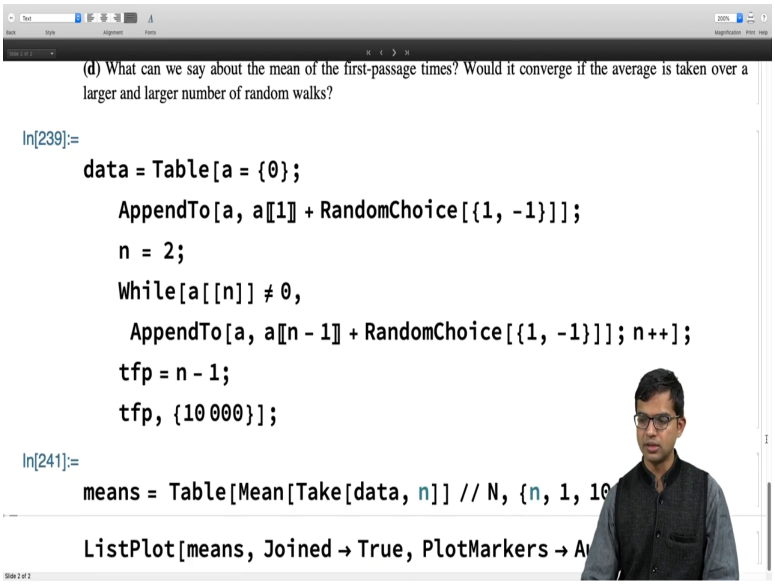


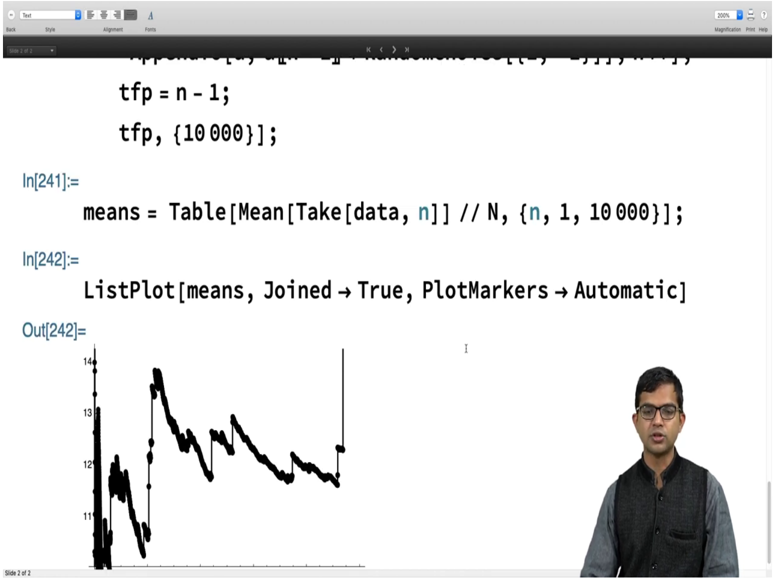
And there is another way of seeing this whole thing. And that is to take, take the mean of you know a different number of these data points. If it is a quantity that is going to converge, then the more data points the better is the mean and it is going to go to some converging value. But you see now actually that in this case it does not converge.

So, what I do is I will generate only one such table with 10,000 numbers in it, it is just one step and in the second step what I will use is, I will use this thing called take, there is a function called take. Which allows me to take the first n of these numbers and take the mean of them. And then I will, I will progressively increase *n* from 1 all the way up to 10,000.

(Refer Slide Time: 17:58)



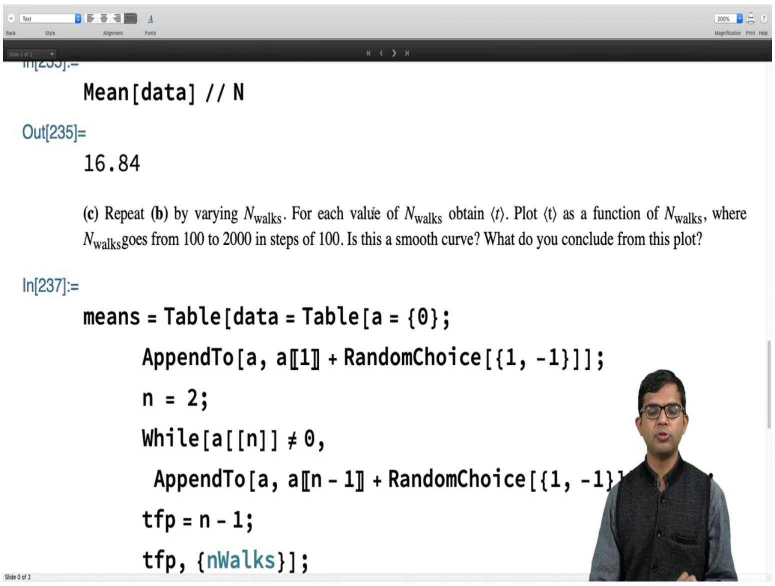


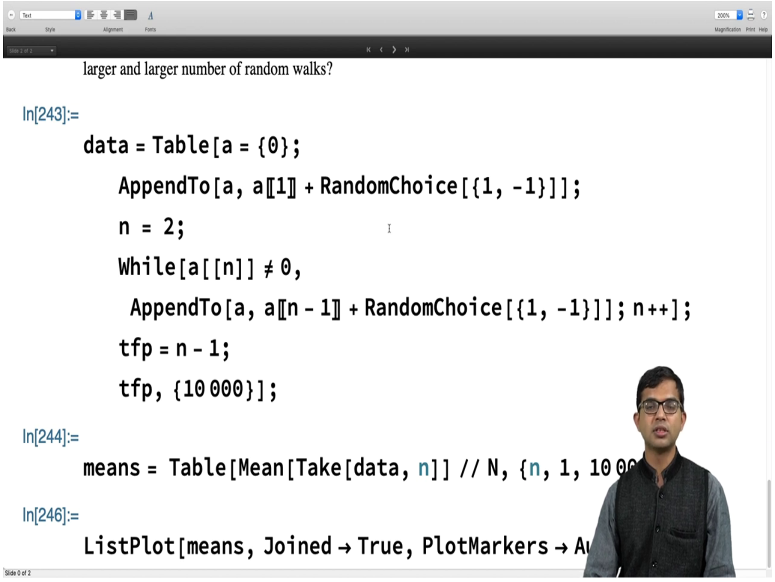


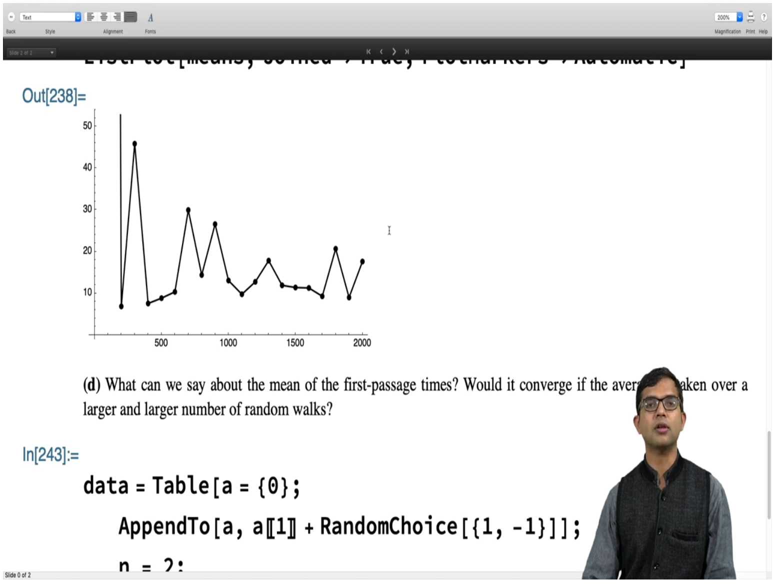
So, there you see it did this and I do not really want to show this data so will suppress this and then if I plot this then you see that it is a wildly fluctuating quantity. If I generate this quantity whole thing again and then if I do this stuff again and then plot this again, once again you see that there is no simple method to it.

If it were a well converging thing you, you would basically get the same kind of graph every time you did it. And so this is what you have to explore and find out.

(Refer Slide Time: 18:36)







So, the final overall take home message from this discussion is that with probability 1, there is going to be first passage to the origin or to any point that you choose. But the average time taken for first passage is actually infinity. And that is why this quantity is not a convergent quantity, this mean. So, this is something that you should play with and try to solve all these bits for yourself A, B, C, D for yourself and then you will see what is going on underneath it. Okay. Thank you.