



IIT Madras

ONLINE DEGREE

Computational Thinking
Prof. Madhavan Mukund
Prof. G. Venkatesh
Department of Computer Science
Chennai Mathematical Institute
Indian Institute of Technology, Madras

Lecture – 4.1
Concept of nested iterations using the birthday paradox (Naive approach)

So, you know there is this famous birthday paradox.

Yeah.

Yeah, it says that I think, if you have some very small number of people; maybe like 25 or something, then there is a better than 50 percent chance that two people have the same month and day of birth, it is very surprising because there are 365 days in a year, but even with a group.

If you have a class of 25; years thing or 23, 24 something.

Yeah, some.

Some class.

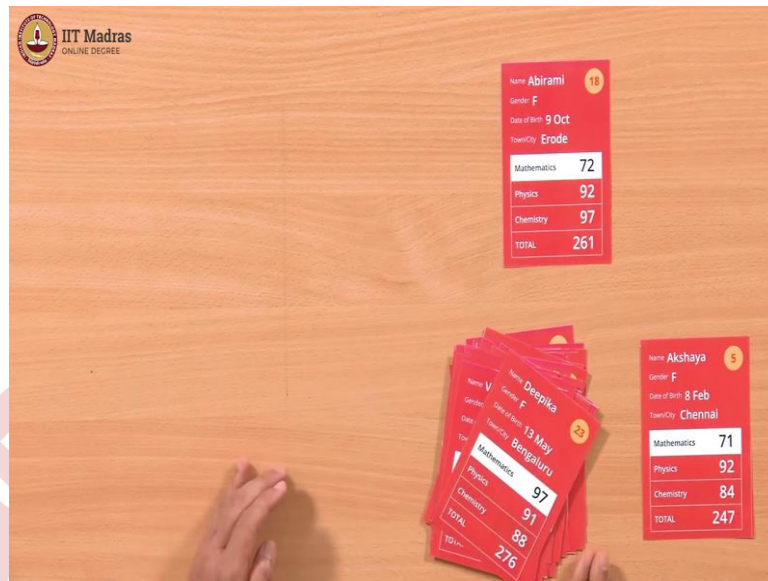
Some small number, you can actually.

Then, they will be; there is a high probability, more than 50 percent probability.

That two.

That two of them will have the same birthday, these are 30 cards.

(Refer Slide Time: 00:47)



So, let us check whether it happens or not. So, how will.

Two people, where you going to check whether two people.

Two people have the same.

Same birthday.

Same birth date in term not.

In same.

Yeah, whatever we have here which is a month and date.

Month and date. Ok.

So, shall we try that?

Right.

So, now we have to first.

I don't.

Seems unbelievable because there are 365 different days, possible or 366.

And we only need to pick two, no.

Yeah and it is very remarkable that this might happen with such a small group; so let us see.
So, what we have to do is now check for say the first card Abhirami.

So, how do you do it? I mean so.

So, we have.

We go through iterate.

Yeah.

As usual, we will iterate.

Yes.

Go through.

So, for every card.

So, Abhirami has 9th October.

So, we have to check whether anybody else has.

9th October has somebody else has got.

As 9th October right and then we have to do it for the.

(Refer Time: 01:34).

Next person and so on.

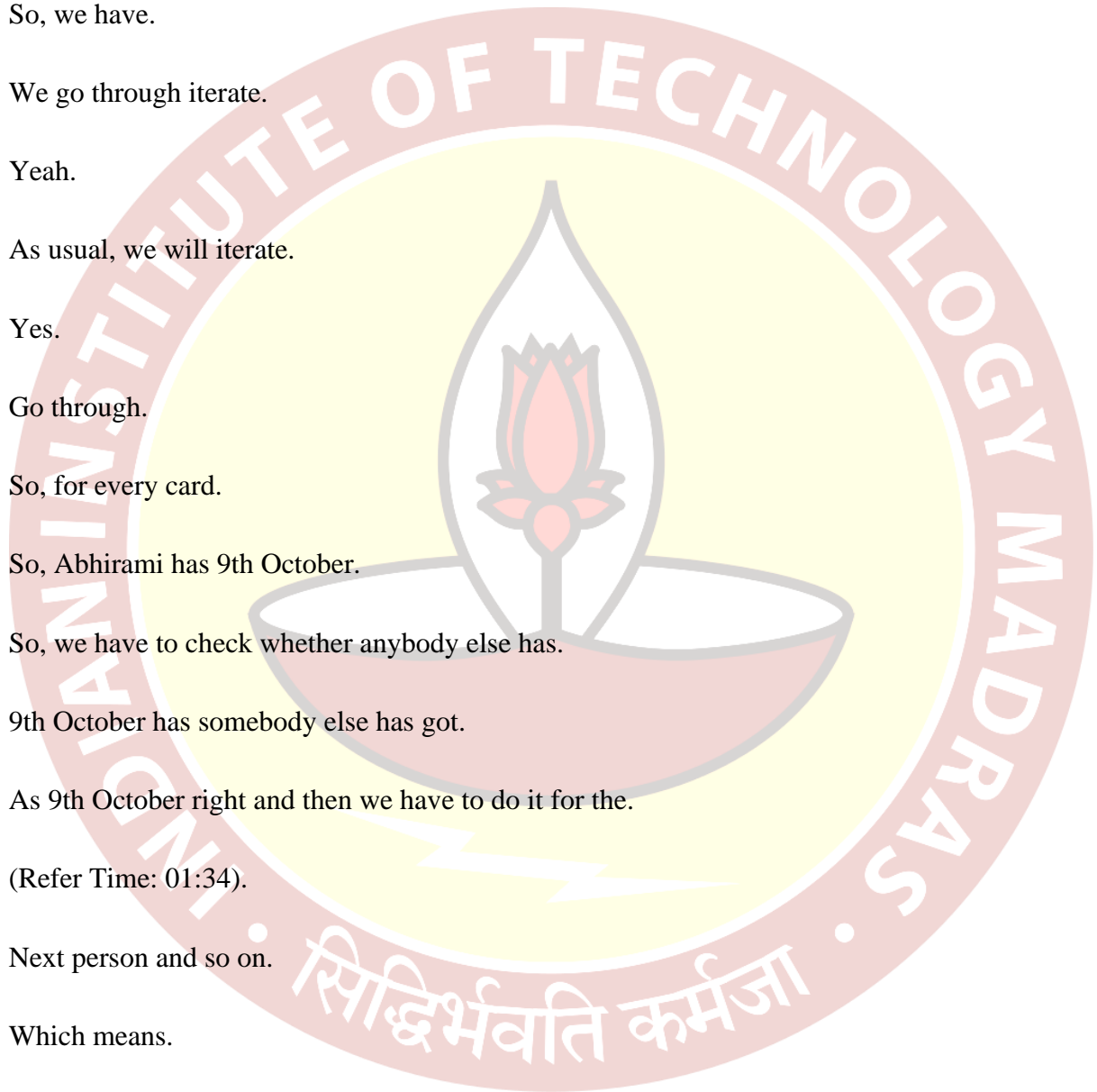
Which means.

So, we.

So, I guess; say I take Abhirami and.

Yeah.

I have seen Abhirami.



Yes.

So, I put it in the second list.

But now we are.

Abhirami have to do something else.

Now, we have to look at everybody else and see if anyone else is 9th October; otherwise we know that Abhirami is not one of them.

So which means you go through this list.

Yes.

The list; which you have not seen actually.

Yeah. So, we have not seen these yet, but we want to check them with Abhirami first and then we will check them on their own.

Ok, so.

So, let us.

So, let us check 9th October, we have picked.

Yeah.

So, one by one; so again, so I keep it here, this is the first.

Card that.

First card is; in the first iteration.

The first card that we are checking ok.

Card.

So, it is.

This is the first card that you are checking the second iteration, not second iteration right?

No.

It is like.

No, this is an iteration; within an iteration.

Within that iteration. So, for every.

Oh.

Every card.

We have actually.

Yeah, so for.

One.

Every student we have to.

Outer iteration.

Check every yeah, for every student; we have to check every other student.

So, we have an outer iteration which is this Abhirami, we have started with.

Yeah.

And now we are starting a?

Inner iteration.

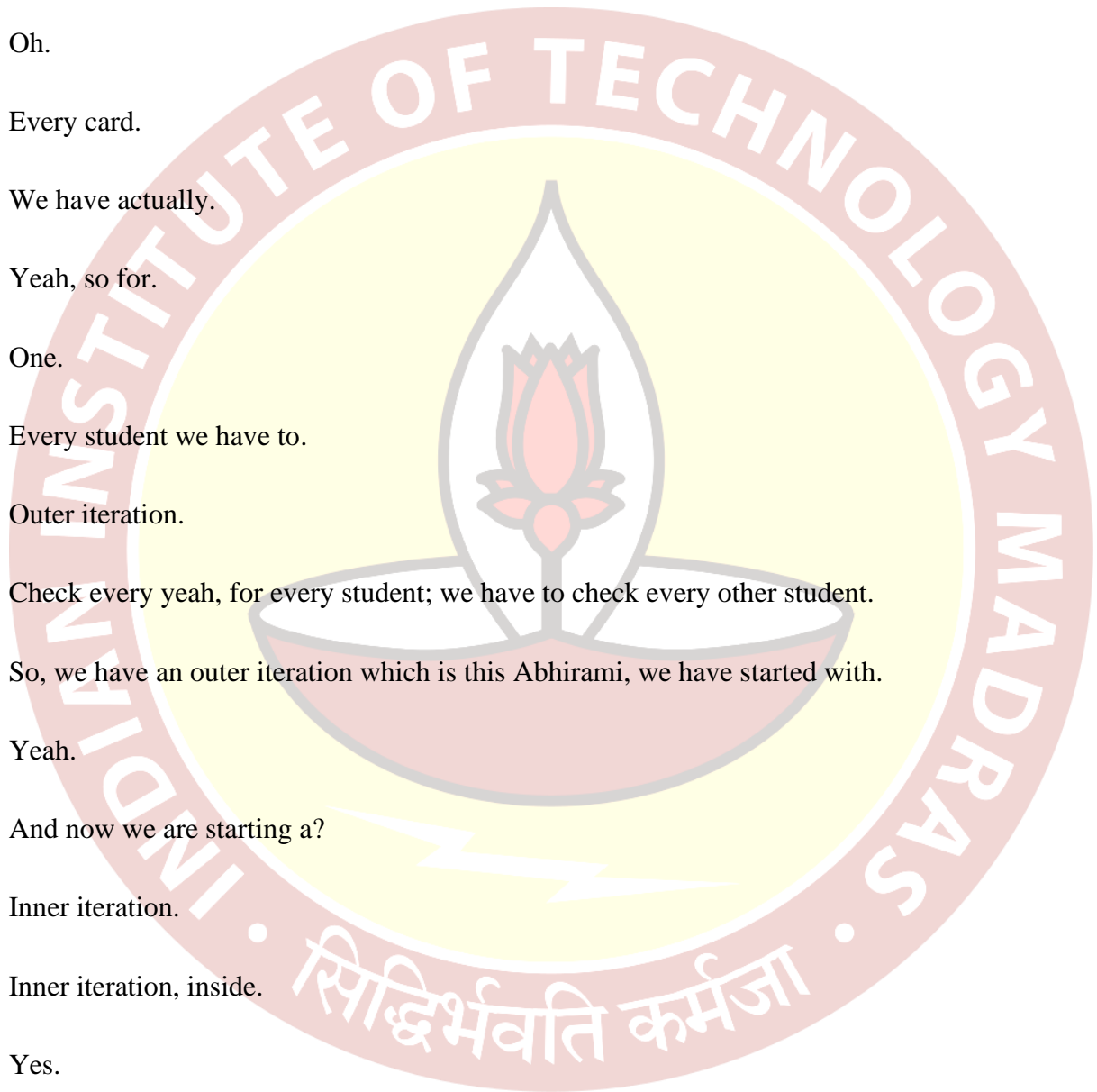
Inner iteration, inside.

Yes.

Right?

Yeah.

So, again it is Abhirami now we are checking. So, again this iteration also we will do same thing.



Yeah.

We will move it to another pile.

Correct.

So, Deepika moves to the second pile.

Yeah, so this is not there.

Not the case.

This is not the case, 30; we are looking for 9th October, so 30th December.

(Refer Time: 02:53).

13th October, so the month is the same, but not the date.

Ok.

26 December, 15th March, 14th December.

4th March.

4th March.

6th December.

6th December.

23rd, 10th September.

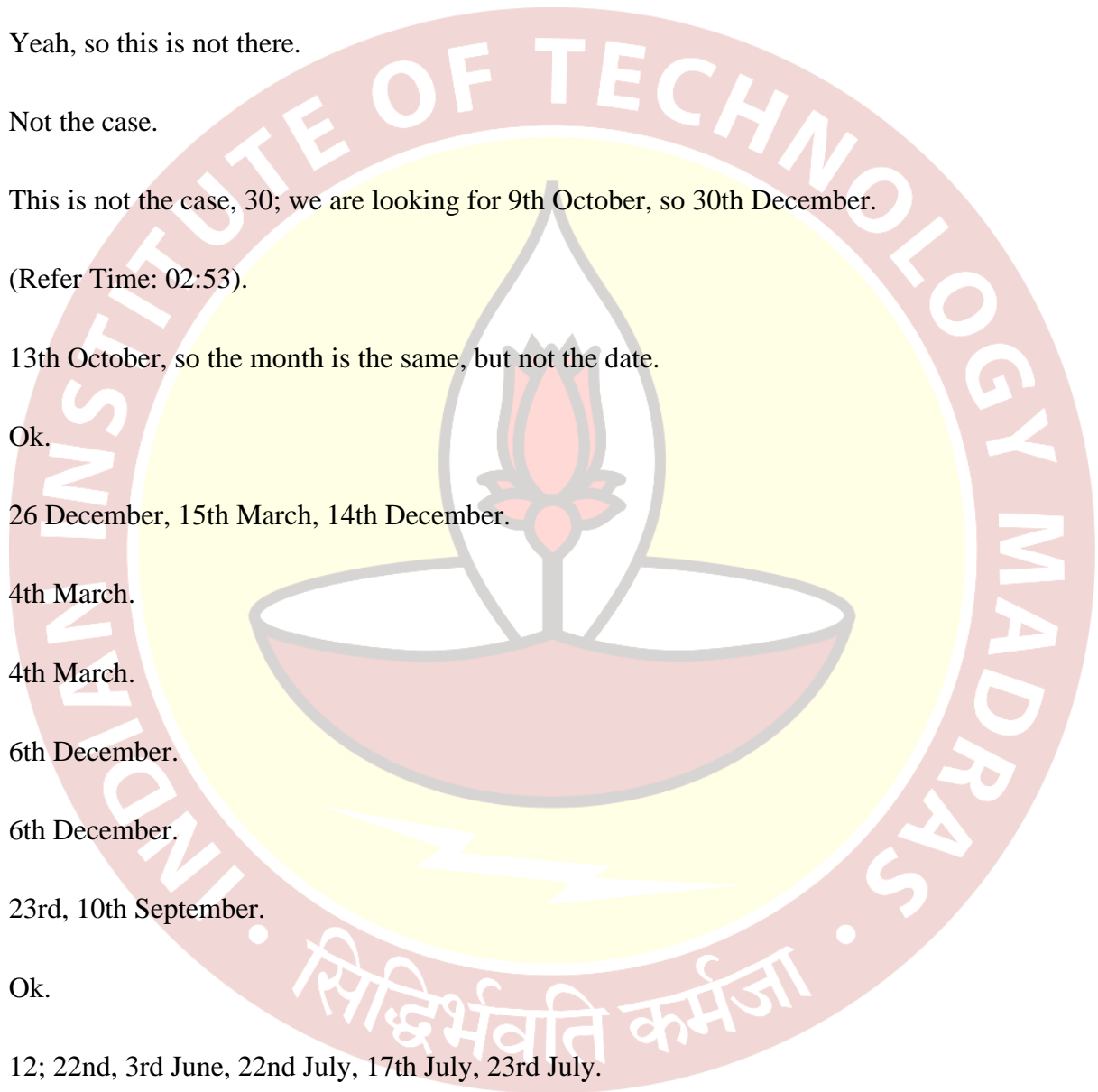
Ok.

12; 22nd, 3rd June, 22nd July, 17th July, 23rd July.

Ok.

17th November. So, we just keep checking and if we see the same date, we will stop.

You are done, you are done actually.



But, right now; so far we have not done that.

So.

So, we keep going until we run out of cards; we finish the iteration.

Ok.

And in this case, we finish the iteration and yeah though we found an October, we did not find (Refer Time: 03:34).

So, 9th October is.

Is not.

Not a date where there are two people with the same birth date.

Correct.

That does not mean that there (Refer Time: 03:40) could not be.

Yeah, because there could be within this there could be two different.

So, what do I do now? So, in the first; the outer iteration, I started the outer iteration.

Yeah.

Picks out Abhirami.

So, we know that Abhirami.

And then I went through an inner iteration where I looked at all the other cards.

Yes.

Right, so I am done with looking at all the other cards.

Yeah.

Now, what do I do now?

Now, we have to do the say for example, for Akshaya, but we have.

So, I have to take; the outer iteration now moves to the next, next card.

Correct, correct and we have to keep track that we have already seen Abhirami in the outer iteration. So, maybe we should create a second stack like this may be.

Oh.

Because see.

Because for Akshaya, I have to compare Abhirami again.

Yeah, for Akshaya we will have to compare these cards and those cards.

Oh, I see.

Because we will have to compare Akshaya with every card.

Every card ok; so, we take out Akshaya, now this is a outer iteration.

Outer iteration.

Alright, now; so for Akshaya, now I have to compare.

First, that one.

It is not, alright ok.

So, we will move it there.

Yeah.

Just to say that we have done; so, we have finished iteration of those guys.

Ok.

Now, we have to iteration of these guy.

These once.

Alright.

So, we are looking for 8th February.

Yeah.

So, this is 28th February which is close, but not good.

Ok.

5th May, 6th May, 16th May, 15th March, 17th September, 14th January, 4th January, 12th, 30th, so if we do not see February; we can quickly go through them.

Yeah.

Like.

Now, this is going to take long time.

Yeah.

Because for every card, we are checking every other card.

Correct.

So, that is like 30 cards and then other 30 cards, 30 into 30.

So, 900 cards.

900 times, 900 times we are doing this, yes.

Oh my God.

So, this is going to take a long time, but ok; so now.

Ok.

So.

Akshaya does not have this; so Akshaya moves into the pile with Abhirami saying outer iteration is done.

But, here.

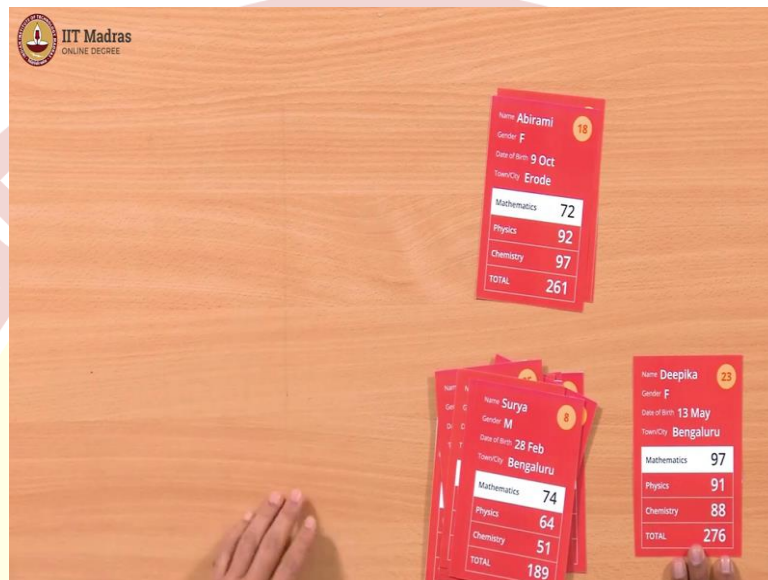
But done.

We are start again.

Yeah.

Right.

(Refer Slide Time: 05:23)



Now, we start again with the next card here, which is Deepika. So, now we are checking for 13th of May.

13th May.

So, then we will say that; first we will go through this.

Yeah.

So, Akshaya is not there, Abhirami is not there.

Alright.

Ok.

Then, we have to.

Then, we have to go to through this. So, again we are looking for.

So, we are taking one card, checking that card against all those that we have seen.

And then all those against we have not seen.

Yeah.

Right.

And then finally, if we finish then we mark this as seen.

And.

So, that we do not (Refer Time: 05:46) any finish.

Alright, I (Refer Time: 05:48); so we.

So, 13th May.

Yeah, no luck so far. So, may be this thing is wrong; may be what they say about these 25 people having a same birth date.

Lot of May's, by the way I notice lot of May's.

Yeah.

But, still we.

But, we are not getting any.

Yeah.

Bad luck.

Ok.

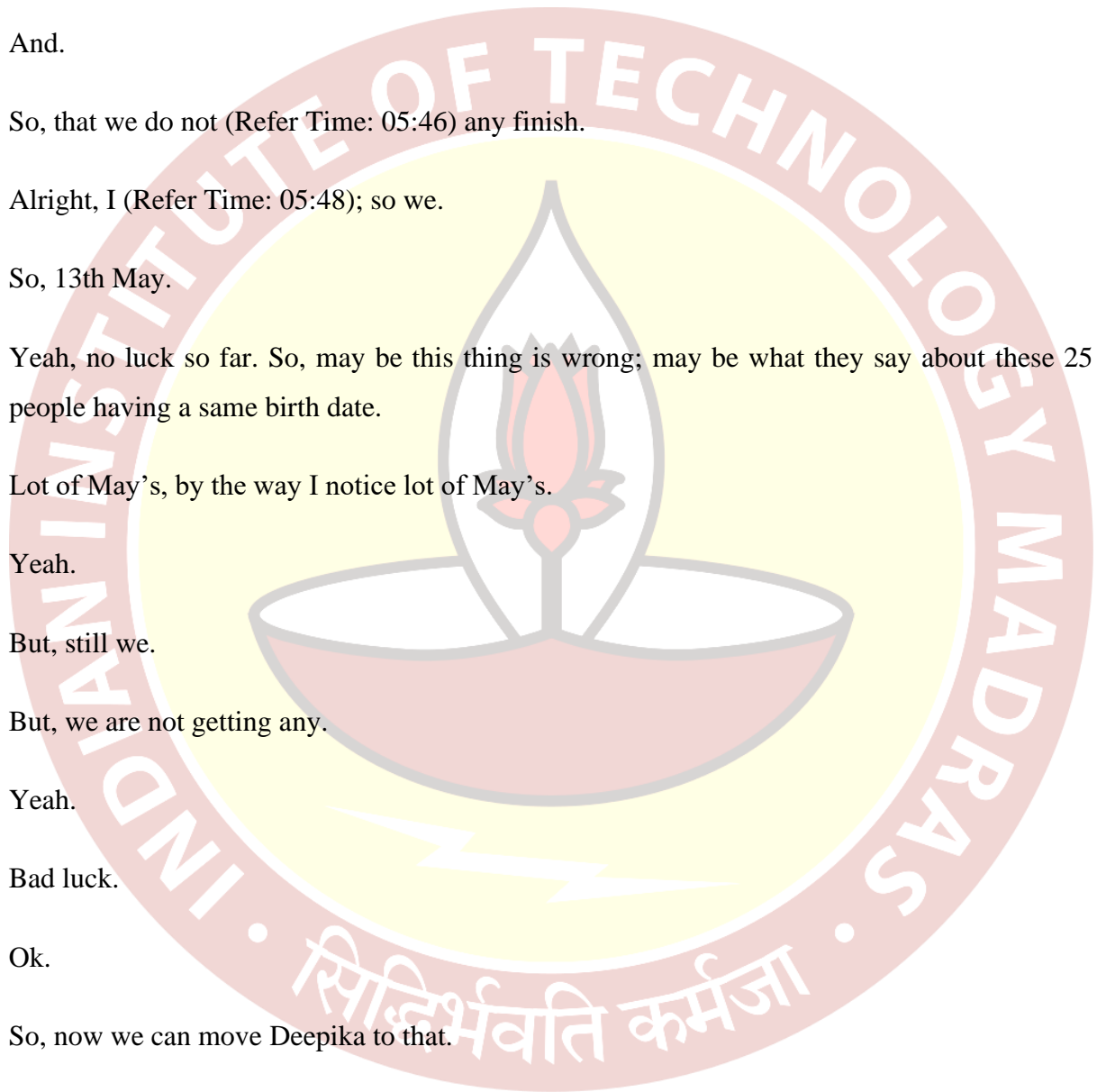
So, now we can move Deepika to that.

So, Deepika and then you go like this.

Yeah. So, now, this is again.

This is the pile that we have seen; this is the pile we have not seen.

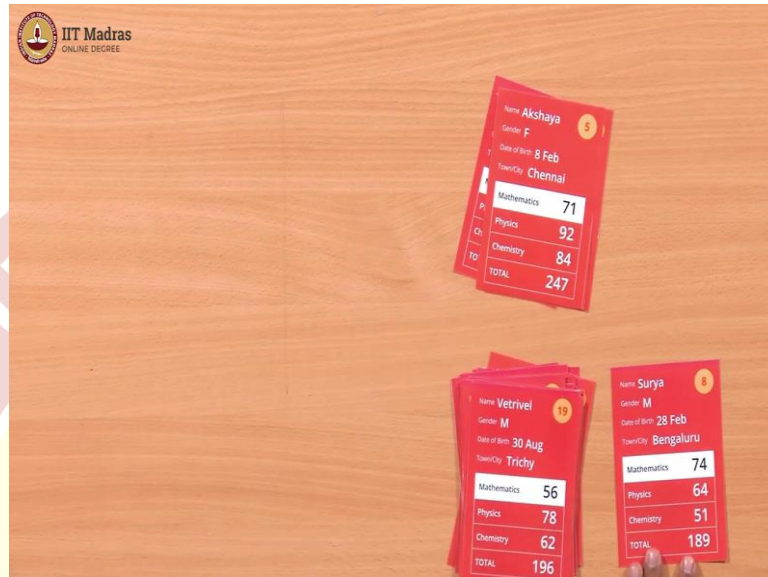
Yeah.



Ok.

So, again we could see (Refer Time: 06:15); Surya.

(Refer Slide Time: 06:15)



So, now again we have to check.

Yeah.

Where the pile we have seen.

Yeah,.

So, 1, 2 and 3, alright now ok.

Now, we have to check for the pile; we have not seen.

5 February right; so, I do not even see a February coming.

So, this is; this nested iteration, one iteration, within iteration creates so many.

Yeah, it takes a lot of time so.

Yeah.

That is something that we have to try and avoid, if we can.

We have to care full.

Yeah, if we have can avoid it; we should.

Can we avoid this? This doesn't look like we can avoid this, right?

Yeah, I do not see it how to avoid.

Yeah, ok.

Ok.

Bad luck.

So, no; so again back.

Back.

Go here.

Yeah.

30th August.

(Refer Slide Time: 06:57)



The image shows three student result cards for the IIT Madras Online Degree program. The cards are red with white text and are placed on a wooden surface. A hand is visible on the right side of the cards.

Name	Gender	Date of Birth	Town/City	Mathematics	Physics	Chemistry	TOTAL
Surya	M	28 Feb	Bengaluru	74	64	51	189
Naveen	M	13 Oct	Vellore	72	66	81	219
Tauseef	M	30 Dec	Trichy	87	86	43	216
Vetrivel	M	30 Aug	Trichy	56	78	62	196

Yeah, but I can see one thing which we can reduce.

See, these guys. We have already checked, against everybody right because we have been through full iteration for all of them.

Yeah.

So, in particular we know that this does not match any of this.

So, come; we have compared 28th February, against all the cards.

All the cards here and.

So, then you take a card from here and compare it with 28th February.

Yeah, exactly.

(Refer Time: 07:26) we take.

So, when we do these new cards; we do not have to go back to.

Because a equal to b and b equal to a is a same check.

Correct.

Isn't it?

So, at least we can save looking at those.

So, we do not need to look at the.

So, each nested iteration will.

Ok.

It becomes.

So, that is something.

Yeah, so let us try that.

It will get smaller each time.

Yeah, it will get smaller each time.

So, instead of doing 30 into 30; now you are doing, first time we did 30, next time you will do 29.

Yeah, so, little bit of saving.

28.

Yeah, little bit of saving.

Here, 27 and as we go further, further; it will become much faster no.

Yeah, hopefully hopefully.

Alright, these all good.

Little bit of saving, yeah.

So, now.

Let us see.

30th August, December, October, December, March, December, March, December, March, September, September, June, July, July, July, November; not even seeing August, April, January, January, January, September.

It is probably not true yaar.

March, May, May, May.

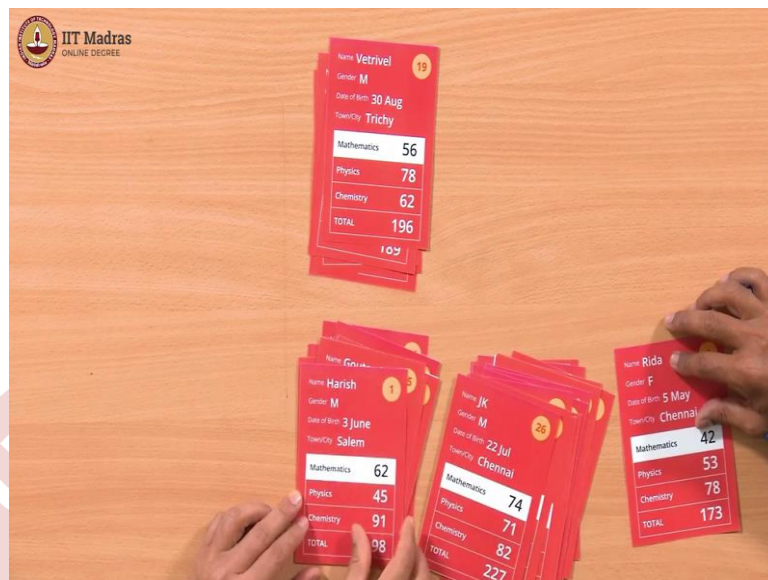
So, no August ok.

Yeah.

So, this is (Refer Time: 08:22).

So, now the good thing is that those guys, we never (Refer Time: 08:24); we will have to just look at them.

(Refer Slide Time: 08:25)



So, now, we have 5th of May; 5th of May; very close, 6th of May, 16th of May, 15th of March, 17th of September, April, November, November, July, July, July, June, September, September.

So, it does not look like the order.

March.

Matters, you know because you are shuffling it as we go longer.

Yeah.

Shuffling, the whole thing.

Yeah.

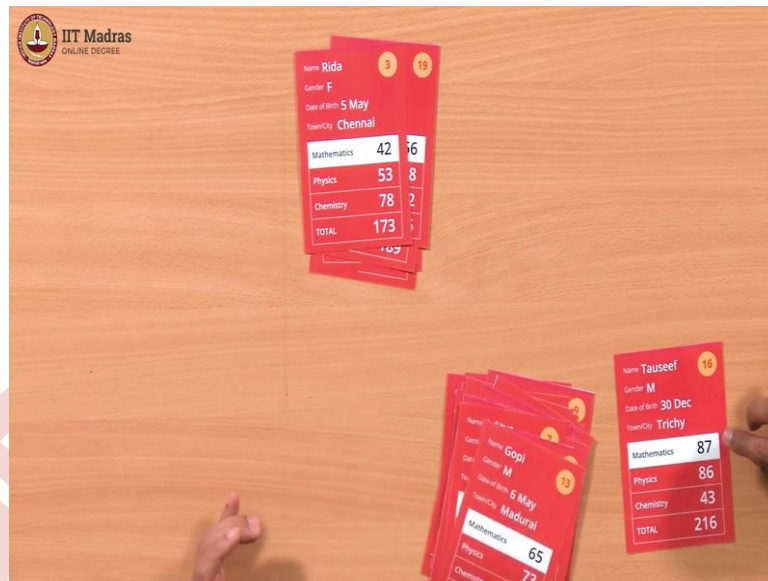
Does not matter, right?

Does not matter because we are just looking for two which are equal at this point.

So, this guy is gone alright.

Yes, so now we have December.

(Refer Slide Time: 08:55)



October, 26th of.

December.

Ok.

March, December; now December yeah.

14th.

(Refer Time: 09:03) here.

March 6th, so lot of Decembers, but nothing is close.

Yeah.

September, September, June, July, July, July.

November

November, April, January, January.

There are lot of Januarys.

January, September, March, May.

May also; lot of.

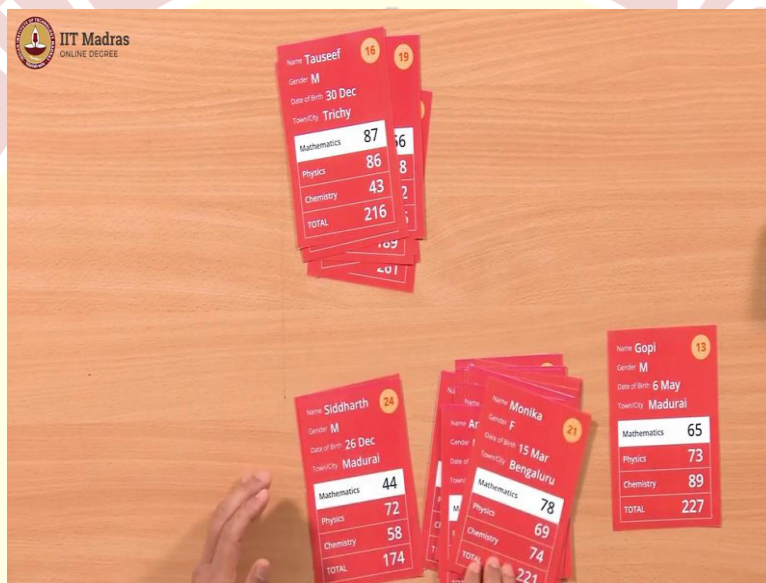
May.

May's.

That is gone.

May, we have luck now; 6th May because 6th May is our lucky guy.

(Refer Slide Time: 09:27)



16th, 15th March, September, January, January, January, April, November, November, July, July, July.

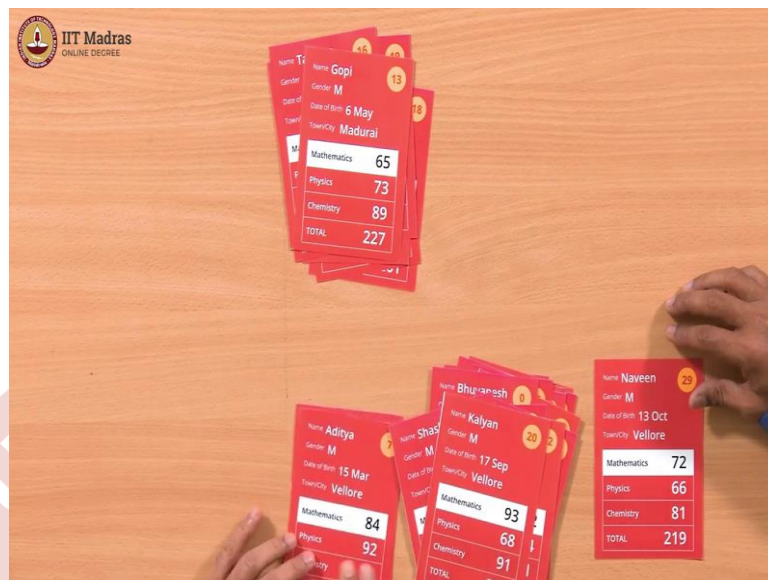
July also lots are.

June, September, September, March, December, March, December, March.

December, October.

No luck ok.

(Refer Slide Time: 09:59)



October 13th, December, March, December, March, December, March, September, September, June, July, July, July, November, November, April, January, January, January, September, March, May.

This guy (Refer Time: 10:24).

So, this is why I think you know when we do this algorithms, when we try to develop programs.

hm

We got to think through how we are doing it right because this business of (Refer Time: 10:33).

Just blindly doing something which may be correct; I mean it is correct, but if we do not think.

It is very useful right.

Think about what I am doing.

I think better and better ways of doing these things ok.

So, May, March; so we are looking for 16th May.

(Refer Slide Time: 10:43)



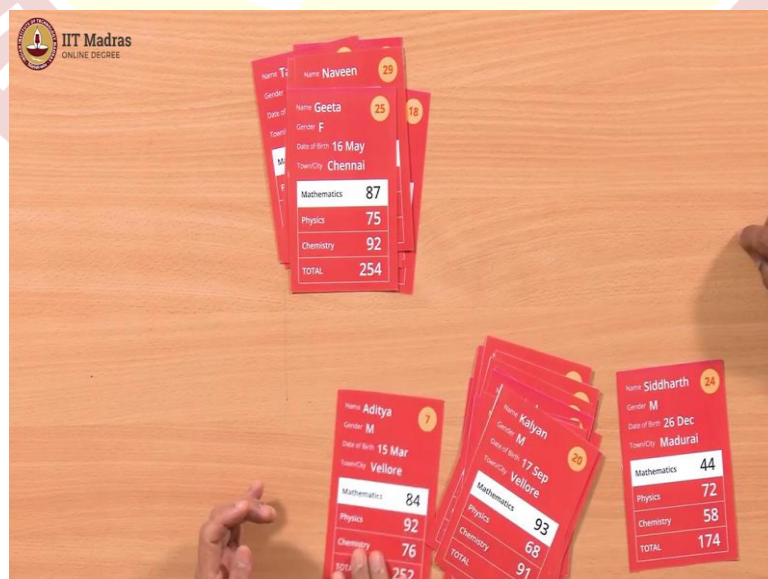
16th May, so March, September, January, January, January, April, November, November, July, July, July, June, September, September, March, December, March, December, March, December.

Ok, it is gone.

It is gone.

So, we are kind of half way through; I think I hope, it goes faster now.

(Refer Slide Time: 11:11)



December, March, December.

December, ok.

But, bad luck; March, again December; now March, September, September; at least each round is becoming little shorter.

Yeah.

Yeah, exactly.

June, July, July, July, November, November, April, January, January, January, September, March.

No luck.

(Refer Slide Time: 11:37)



Ok, so now, let us try; 15th of March, alright.

September, January.

Lucky guy.

January, well January, April, November, November, July, July, July, June, September, September, March.

No; March.

No, 6th December, 4th March.

4th March.

No, 14th December; yes.

Yes.

So, we found a pair.

Wow.

So, it is amazing.

Wow.

So, it is actually true; I mean, at least what the probability tells us. Actually has come right in this case. So, we did find a pair, but it took a lot of work.

So.

And of course, there could be more pair in this, but.

Yeah, there are more pairs.

But, let us so.

We are happy that we have found one pair.

One pair. So, what did we do? I mean let us just reflect, what we did.

Yeah.

We said we have; this is not possible to do without a loop within a loop, an iteration within an iteration, right?

Yeah. So, I do not think; if we just have these cards.

Because, we just did because you have to compare everything with everything else.

Yes.

So; that means, that; if there are 30 cards, you have to compare 30 cards against, 30 cards of course, we try to.

Yeah.

Not compare it twice.

Yes, we have compared one A with B.

If we have compared A with B, no need to compare B with A again.

Yes.

That way you can reduce a bit, but still kind of you have to compare everything with everything, right?

Yeah.

Now, but an; if you have to compare everything with everything, you cannot do it with one iteration followed by another iteration; like we did earlier.

Yes.

Because one iteration will do some 30.

Things.

Steps, it will do and then this iteration will do another 30 steps.

Correct.

Whether, if you are doing this into this.

Yes.

30 into 30, we have to do 900.

Yeah.

May be, you reduce it to half because you have.

Yeah, because each one you accounting this; it is.

Twice counted.

Yeah, so 900 may become 450 or some; I do not know

Yeah.

Some not small, slightly smaller number, but it is still.

Yes, so you cannot settle it, do it and

you cannot do it in.

(Refer Time: 13:33).

A fixed number of iteration.

Cannot do it; so, it has to be one iteration inside. So, we have now; till now we have not seen this right? We have not seen this.

Yes.

Business of taking one.

Yeah.

Iteration.

Fixing.

(Refer Time: 13:44).

Fixing one thing and then comparing.

Putting another iteration.

Yeah.

So, one iteration sitting inside another iteration; one right. We have not seen this before; now it is complicated because it seems that keep track of things I mean of course, with a small; so we were able to do it.

Yeah.

But keeping in track and when you do this, inner iteration inside and outer iteration; keeping track becomes difficult because when you are doing the outer iteration, you have seen some cards. You have kept it in a different pile.

Yes.

And then you are going to see some cards, now this inner iteration has to go through the whole thing which means it has to go through

in general.

The cards.

That is right; if we do not.

In which case it turns out, basically that we did not have to, but you have to go through the cards that you have already seen, then you have to go through the cards; you have not seen.

Correct, so I agree; so lot of.

Lot of this administration.

Administrative work.

Yes, but.

So, and if you have iteration within iteration, within iteration then.

Yeah, yeah for instance that would be even harder right.

Even harder.

Yeah.

Right and it will take a long time. Because you know, this 30 into 30; 900 will be; 30 into 30 into 30.

Correct.

And 30 which is like you know.

2700.

Oh, my God.

You will have to.

27000.

27000, right.

So, we will never be able to do it manually; we need a computer to do it right.

Yeah.

So, any way; we saw an iteration within an iteration for the first time.

Right.

Are there any other examples of this iteration within iteration that we can think of? Where does it come? When of course, dates; common dates, when does it come and why do we get iteration within a iteration?

Yeah, so whenever we have to do this kind of.

Pair wise.

Pair wise comparison right. So, you want to look at everybody and so for instance, if we looked at our shopping bills. Supposing, we wanted to find out two customers who shop in the same shop; then we would again have to do this right, you would have to.

Have.

Look at.

Have both the customers gone to the same shop; if I have to find out, then first I have to take this customer, find the shop, find the bill.

Yeah.

Which shows, whether the customer is.

(Refer Time: 15:42) even more complicated.

Even more complicated.

Then, from that shop I have to find; want to try that, I mean then that shop find out whether there is another customer.

Yeah, maybe there is a better way to do that is there a better way to

Two customers; who shopped in the same shop.

So, I think that there may be, in that case a slight improvement we could do which is to separate out the bills by shop. Right, we could have all the bills for one shop, all the bills for another shop, all the bills for a third shop. So, we can first do this kind of which we have done before.

If we are done before if we had when we had done (Refer Time: 16:13); separated it by shop.

Separated by some category, like all the people who had got marks between 50 and 60 or something like that.

So, in that case; you just have to count whether there is more than one person.

Then, we have to look in each shop, we have to just look do the same thing, but even within a shop; we have to do that iteration, but for a smaller group.

So, it is one iteration following another iteration, that way?

No, no it will be a first iteration to separate, but within each shop; we have to do a nested iteration right, we have to look for no need; no any two people in that.

Any two.

Yeah.

(Refer Time: 16:38); we have to see whether there are two different names in that.

Yeah, correct.

List.

Correct, yeah at least.

That won't give you a nested iteration to you.

Yes.

Gives you only a.

Yes, so that is an easier problem.

Yeah.

So, in some sense the complexity of this problem you know; so you can we are already seeing that the problems are falling into buckets right, some problems are looking like they are somewhat.

Yes.

Let us; I mean, I am just saying, using the word easier.

In the sense (Refer Time: 16:58).

Because in the sense that one pass.

We can do them in one pass or even if we do not do it one pass; you do one pass, maintain some values and do one more pass.

Yeah.

Second pass and you are able to solve the problem. Whereas some things like this, it did not look like; I mean finding two people have the same date look, I mean at first phase looks like an easier problem to do an.

Yeah.

Three prizes.

Correct, correct.

Three prizes; we did one in pass.

Yes.

Right?

Yeah; essentially because one pass to find the top three and then one pass to find subject wise top three so, that.

So, two pass.

Fixed number of passes.

Two number of passes, two passes we did.

Yeah, but here we have nested iteration.

Yeah.

So, this looks like; while the problem looks simpler than that, that problem looked much more complex in terms of the statement.

Yeah.

But to solve, it was easier.

Yeah.

This problem on the other hand.

Is easy to state.

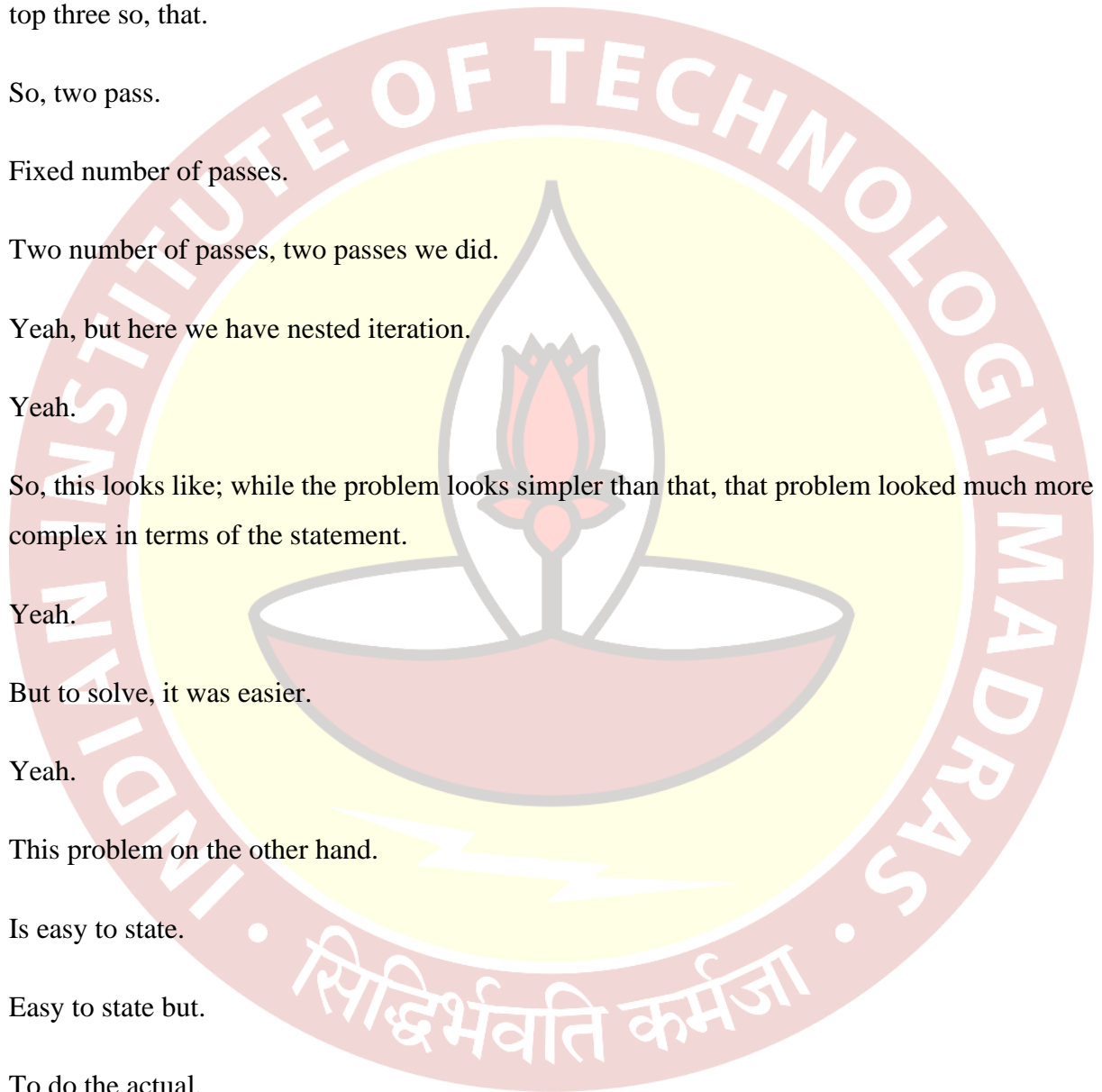
Easy to state but.

To do the actual.

Do the actual things, seemed very hard.

Yeah So, maybe we will come back and see whether we can do something better with this or with other problems and see what makes these problems require this kind of thing.

Yeah.



As we go along.

Ok.

