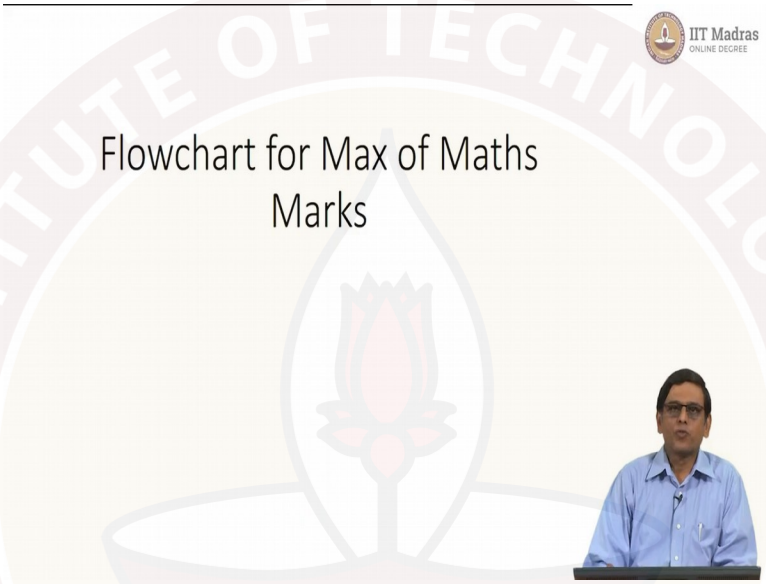
IIT Madras

ONLINE DEGREE

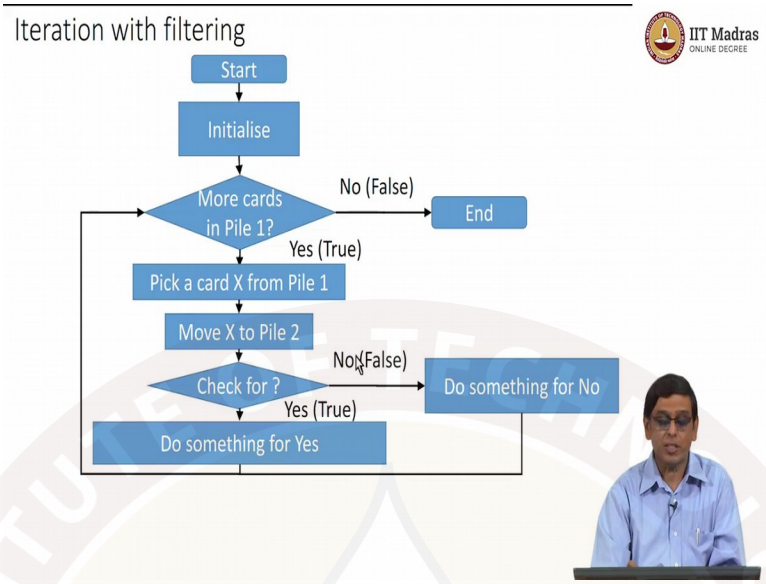
Computational Thinking
Professor Madhavan Mukund
Department of Computer Science
Chennai Mathematical Institute
Professor G Venkatesh
Department of Electrical Engineering
Indian Institute of Technology Madras
Flowchart for Max Marks

(Refer Slide Time: 0:15)



So, we have seen how we can do, we can do a procedure for finding the maximum card which, card which has the maximum marks for maths. Now, what we are trying, what you are going to do in this, in this particular lecture is to see how to turn that procedure into a flowchart. So, as we have seen before the main thing to do basically is to see whether we have a flowchart already that we can modify for this, for this particular problem.

(Refer Slide Time: 0:49)



So, I am going to start with the flowchart for iteration with filtering, some kind of generic flowchart for doing iteration with filtering. So, if you remember the flowchart for iteration was that we start, then we have this box which is basically, remember that rectangular box represents action or a process. So, we have to initialize the iteration, a diamond represents a decision which means that they are checking for something.

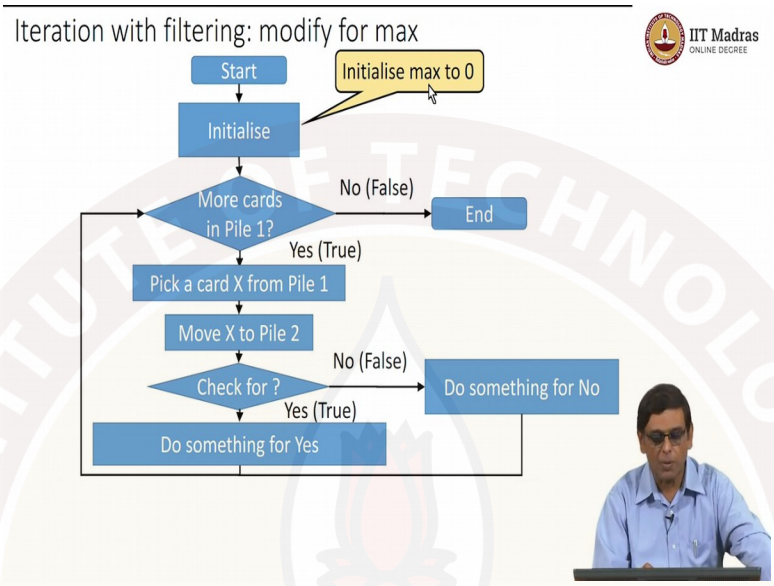
Here we are checking whether there are more cards in the first pile. So, assumption is that when we start, all the cards are organized in a pile which is called pile 1. So, you check whether there are more cards in pile 1. If there are no more cards, then that brings us to the end of the flowchart. So, the start and the end of the flowchart are denoted by these oval boxes like this. And in case there are more cards in pile 1, you pick a card X from pile 1, move it into pile 2 so that you do not visit this card again.

And then you have to check something on this card, so that is where we are doing filtering and based on something that is there on this card you decide whether to do something, in case that something is present, like for example if you are checking for, if you want to do the sum of boys, you check whether the card is a boy or not and which case you add sum to boys of sum and if it is not then it is basically a girl then you add the sum to girl sum, something like that.

So, you do an action for the true condition and you do another action for the false condition. And in both cases you go back, after you finished doing that, you go back to the beginning of the

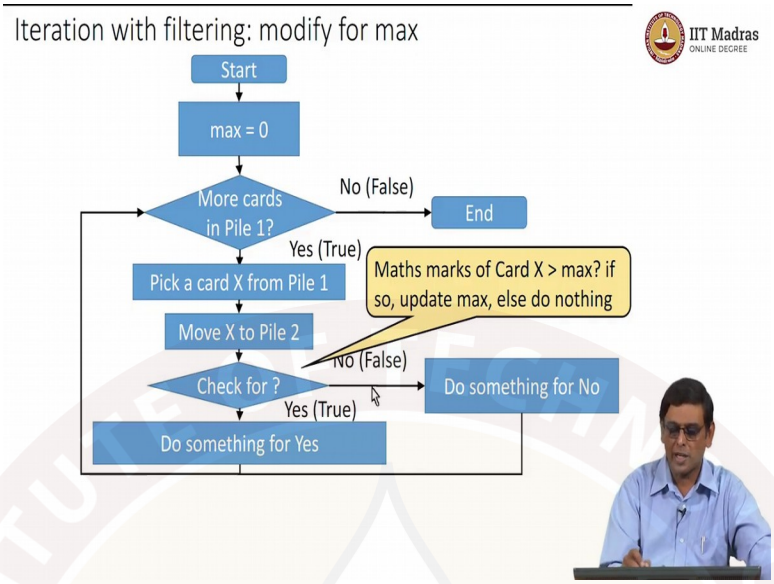
iteration which is basically this check over here. So, we are going to start with this generic flowchart for filtering, and we want to see whether we can modify this generic flowchart to do what we want, which is basically to find the maximum maths marks.

(Refer Slide Time: 2:49)



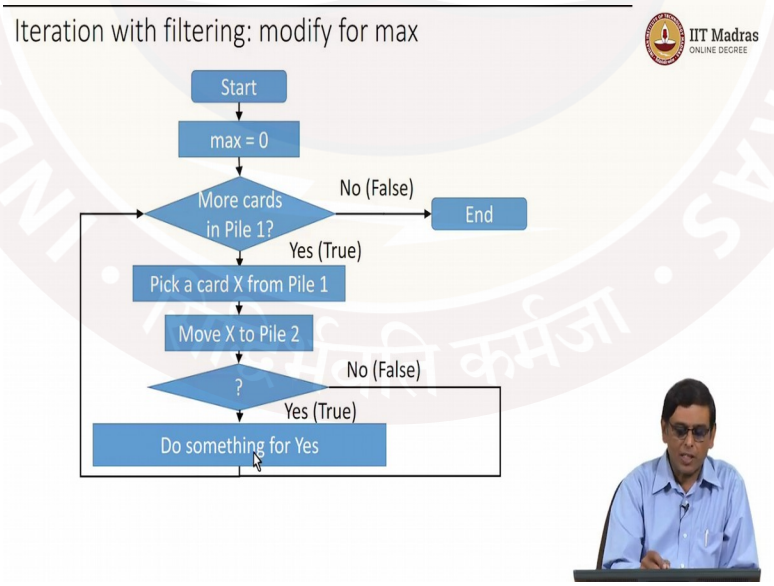
So, if you remember what we did for the maximum maths marks was we had basically to do an initialization and the initialization was to set the max to a suitably low number. So, 0 is a suitably low number because no marks, no maths marks can be below 0, so therefore we set max to 0. And then we go through the same process, somewhere here instead of filtering, which means that we are comparing the card with some constant value. We said let us compare that value of the maths marks on this card to the max variable that we stored.

(Refer Slide Time: 3:26)



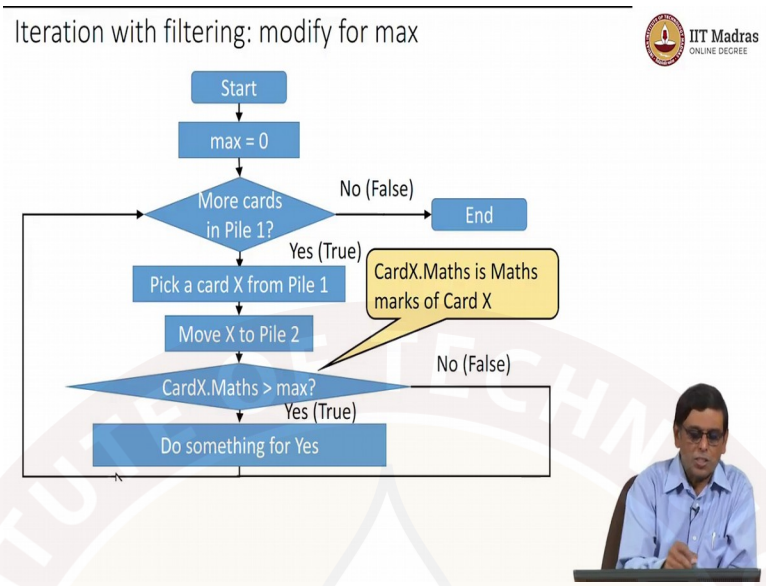
And if the value is, is less than the max marks then there is nothing to do. Then there is nothing to do, on the other hand, if the value of the max of the card X is maths marks is more than max, in that case you update the max value. So, that is the procedure that we have done. So, basically the way to represent nothing to do is to simply remove this box and put this thing like this.

(Refer Slide Time: 3:47)



So, that is what I have shown here. So, basically you are going to check something here and then after that if it is, if that, if the value is less than the max marks then basically you do nothing, you go like this. And if it is greater than max marks then you update the max marks.

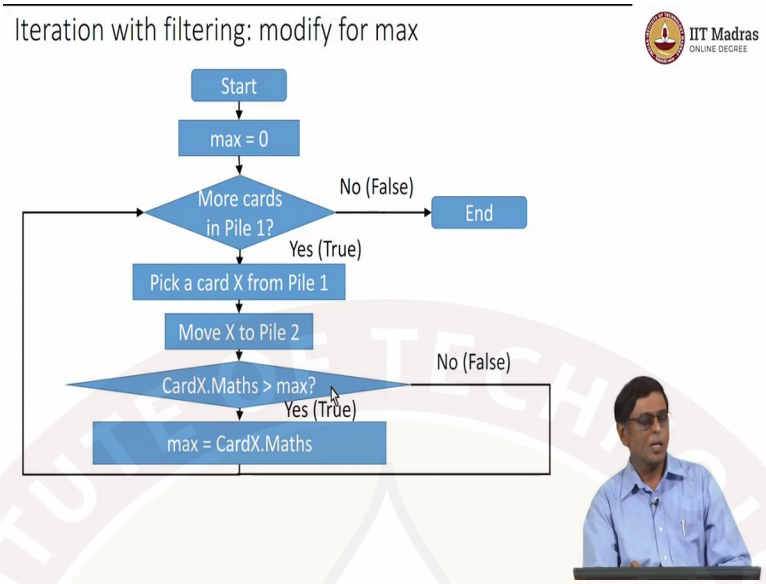
(Refer Slide Time: 4:04)



So, how do we check whether or not the value of the maths? So, I have written cardX.Maths, as it a kind of notation and we will see as we go later when we do this slow code kind of stuff, you will basically that we will introduce all kinds of interesting notation to access fields of data structure. So, in this case I have used basically cardX.Maths to basically extract the maths marks for the card X.

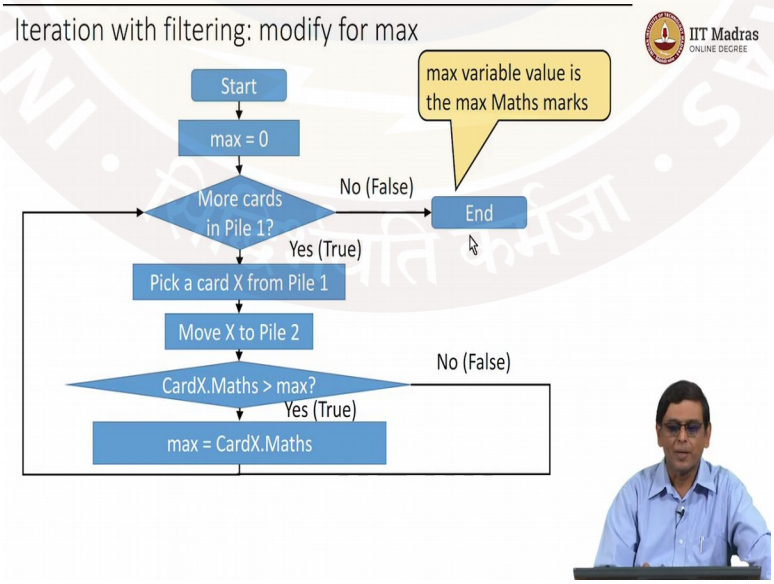
So, cardX.Maths. we are checking whether that is greater than the variable max that we have stored. If it is greater, if it is not greater which means it is either equals to or less than, then there is nothing to do. So, you go back like this, if it is greater, then we have to do something. And what is that something that we have to do, we have to update the value of max.

(Refer Slide Time: 4:53)



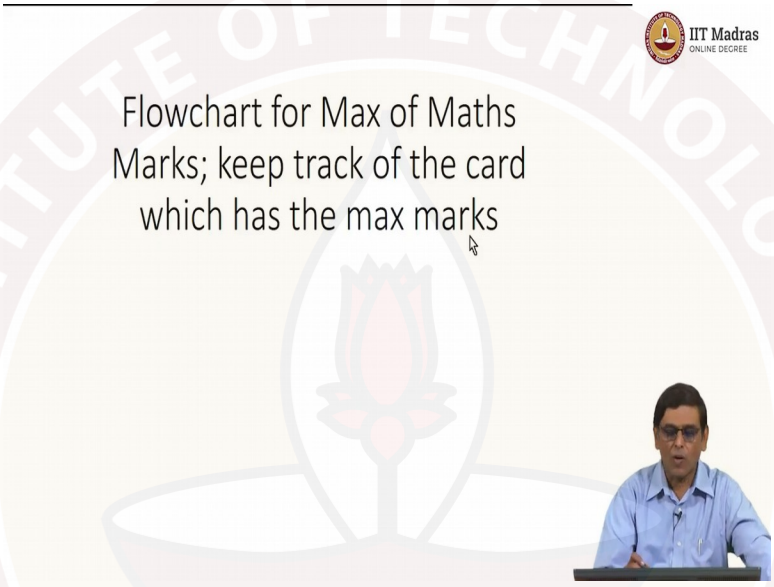
So, the way to update the value of max basically is to set max to the value of the maths marks on card X. So, maths is equals to cardX.Maths. So, the simple modified iteration process with filtering basically, instead of comparing cardX.Maths with some constant value we are comparing it with a max variable and if the value is greater, then we update the max variable and then go back to the beginning of the iteration. It if is not greater there is nothing to do, so you go back to the beginning of the iteration.

(Refer Slide Time: 5:24)



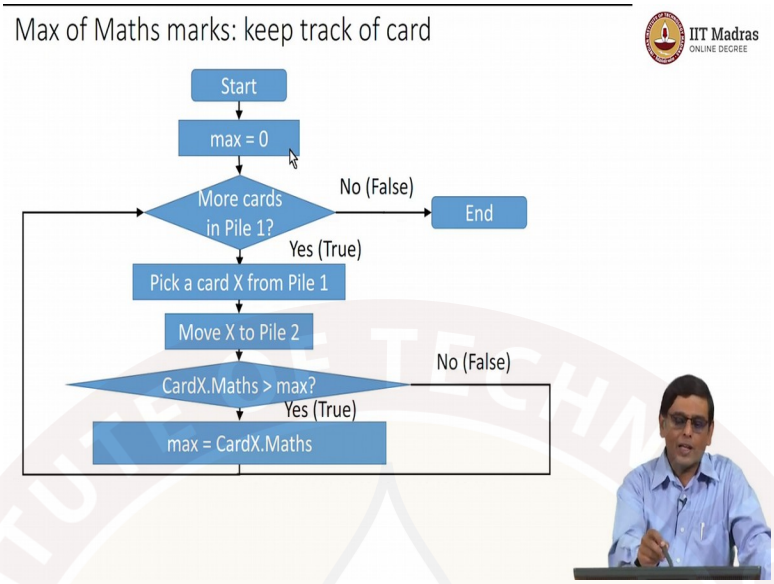
So, it is clear that at the end of this process which means when does the process ends, this procedure ends basically when there are no more cards in pile 1. At this time, you will come out on to this terminal, and at this point when you are at the end of the procedure, the value of the max variable will be carrying the maximum maths marks of the cards that we have seen in the pile 1. So, this is very fairly simple way of modifying the flowchart that we say earlier, the filtering flowchart to find the maximum of maths marks.

(Refer Slide Time: 5:55)



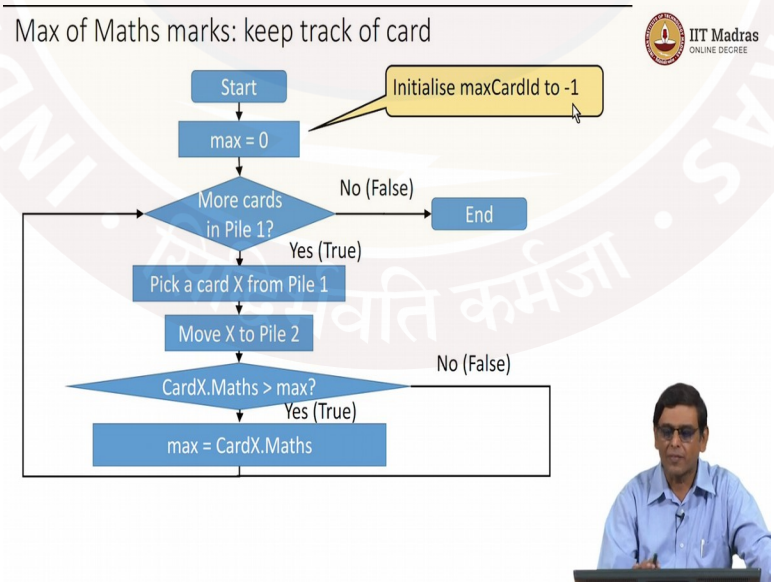
Now, we also discussed basically that not only when we want to find the maximum of maths marks but we may want to keep track of the card which has the maximum marks. So, the next question that we can ask is, what should we do to modify this flowchart so that not only do we find the maximum we can also keep the track of the card on which that maximum was found.

(Refer Slide Time: 6:18)



So, this is the flowchart that we just presented for finding the maximum, so what change do we make to it to keep track of the card also because we can see here that the only thing we are keeping track is a variable called max and there is nothing we are not keeping track of the card itself on which this max was found. So, we have to do something further not only to keep the value, maximum value but to keep track of the card.

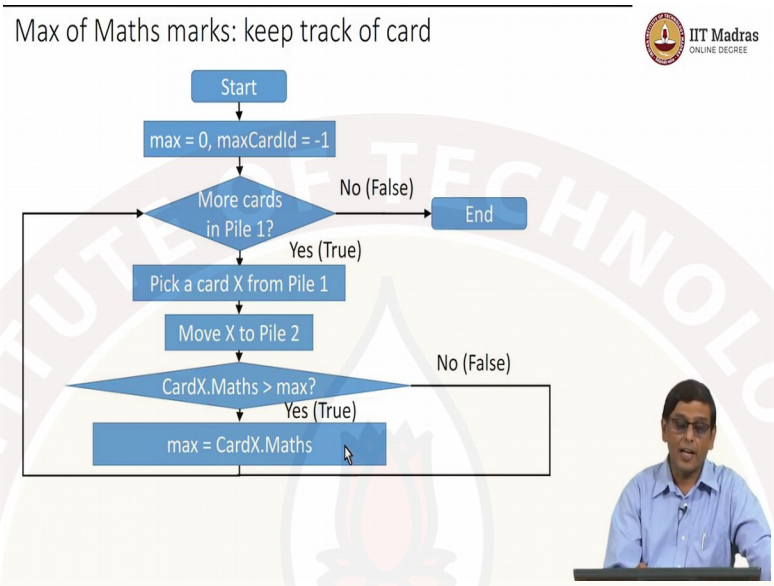
(Refer Slide Time: 6:45)



So, we are going to use another variable like we did call maximum cardId and we have to initialize that to some value which is suitably, it is not present actually something that is which

represents null or nothing is there, I have use minus 1 to represent that. One can use anything else, you can use any other character which is not going to be found as a Id on any of the cards. To represent basically that initially the value is set to some null value.

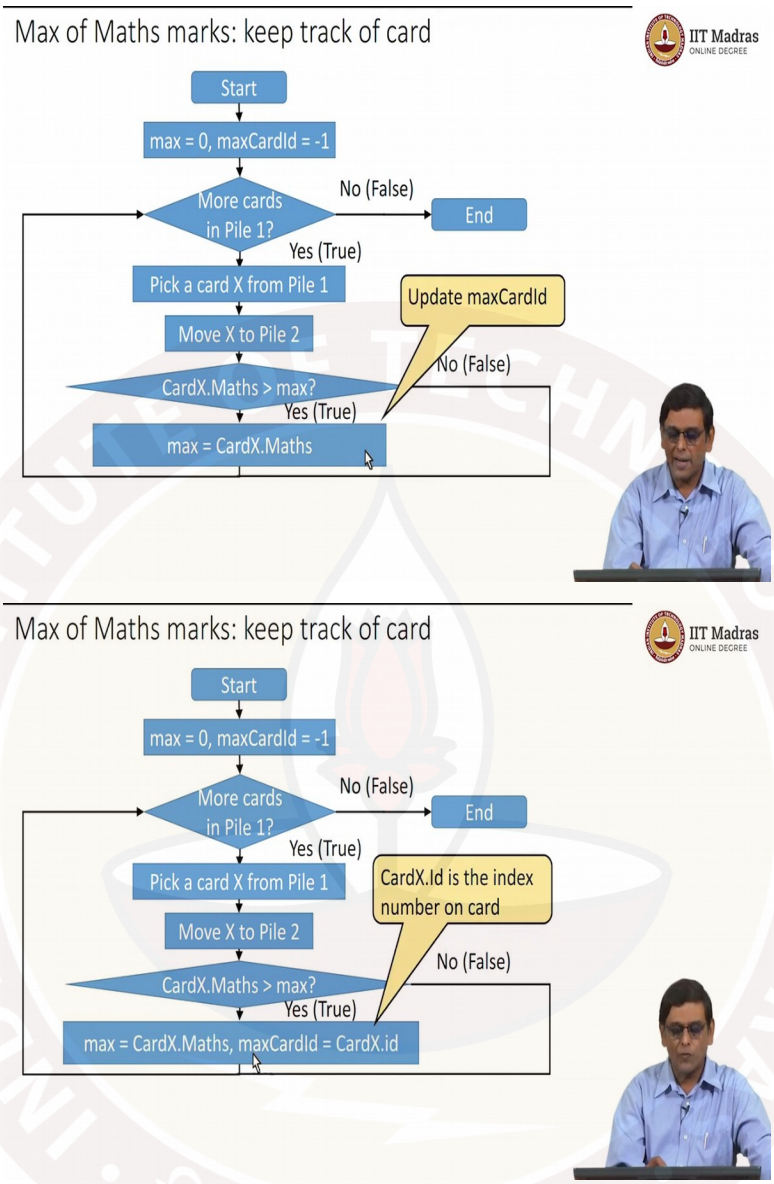
(Refer Slide Time: 7:13)



So, we have set basically the maximum cardId, not only did we initialize the variable max to 0, we have also set maxCardId to minus 1. And you go through the same procedure, now there is no change to be made here, here or here or here because we are not going to actually compare anything with a cardId.

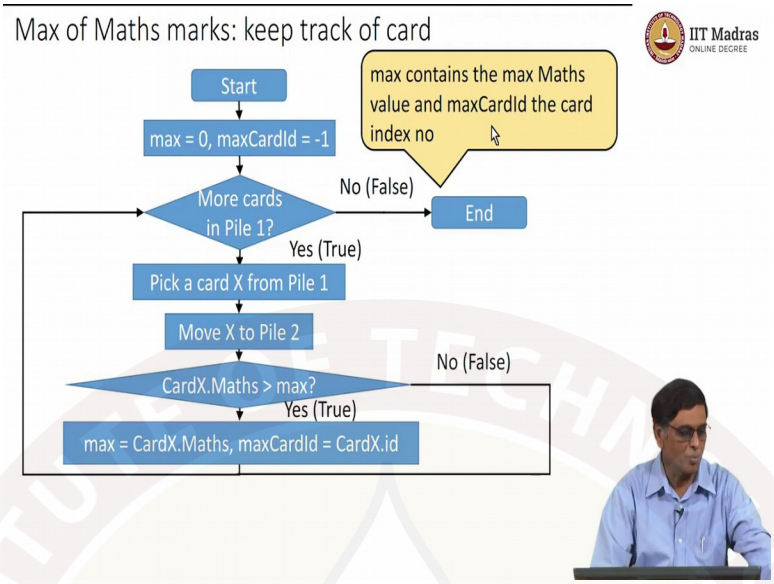
We are only going to compare it with the value of the current maximum value. So, this comparison is also okay, so the only other change we have to make is when we do this update. So, in the event that card X is greater than max, then we not only have to update the value of the max but we also have to update the value of the card on which that max was found.

(Refer Slide Time: 7:56)



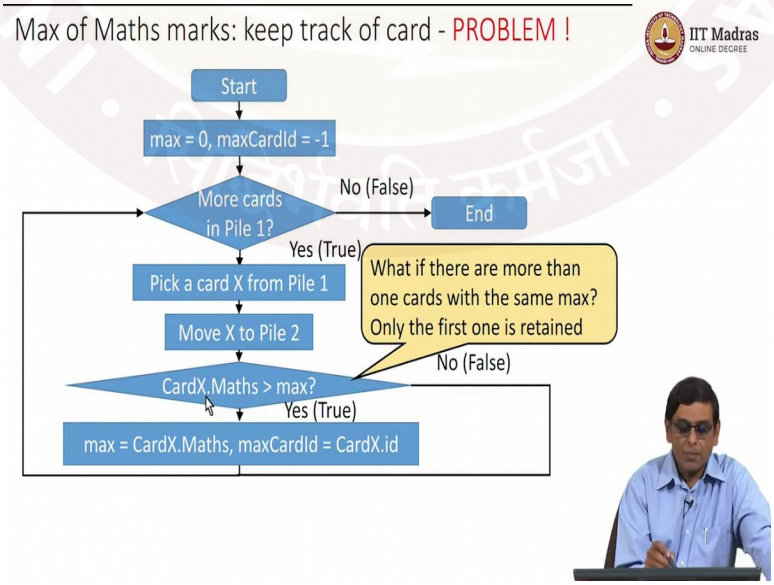
So, we have to add one more, one more update, one more process or one more activity here inside this box which basically is maxCardId equal to CardX.id. So, just like CardX.Maths denoted basically the value of the maths marks on card X, they are using CardX.id to represent the Id or the unique number that we have given to each card. So, maxCardId is set to CardX.id. So, this procedure now, this new procedure now, you will go through this procedure iterate and when we will stop? We will stop when there are no more cards in pile 1.

(Refer Slide Time: 8:33)



So, you will come out at the end here and when you come out at the end, now the value of max variable is the value of the maximum maths, maths marks, maths marks what you have seen so far but also because the keeping track of the card on which that max marks was found, maxCardId will carry the card index number as well. So, if at all we want to find which card contributed to the max marks, we can use the card Id to go and look for that card, extract that card and we will know which card, which person or which individual has got the maximum maths marks.

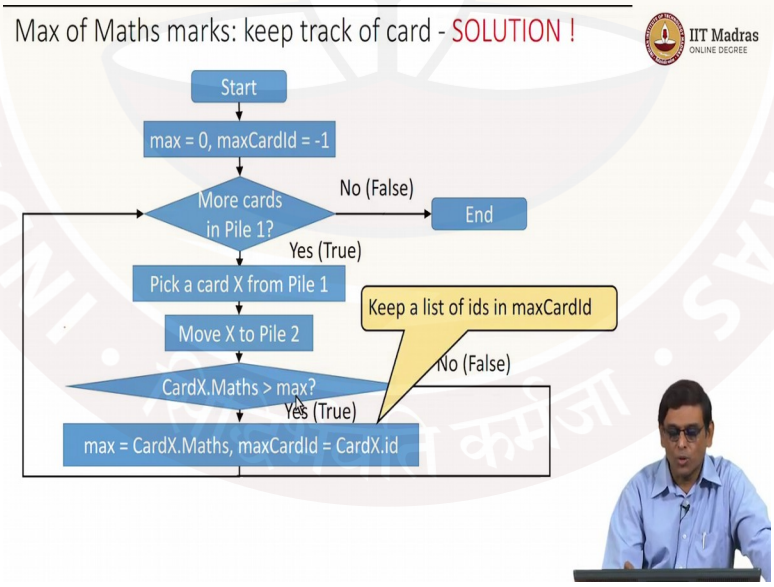
(Refer Slide Time: 9:06)

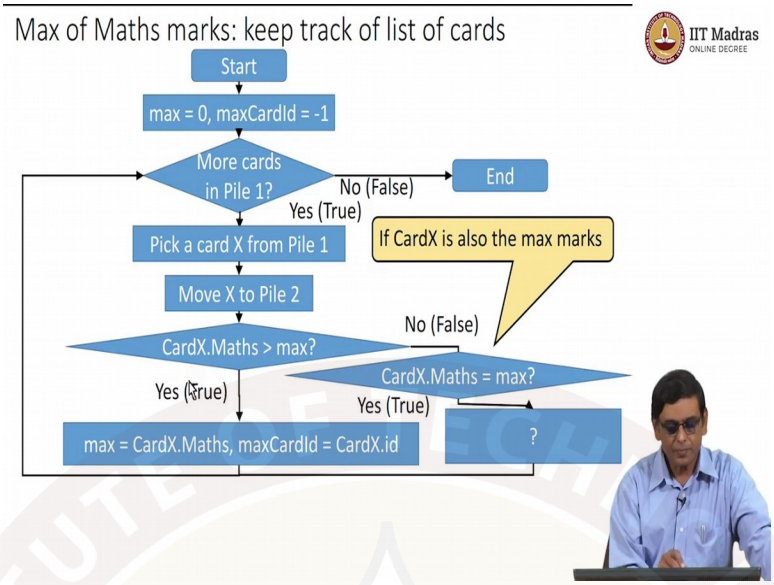


Now, there is a problem in this, right, as we discovered when we did with hand, which is that you could have for example two, two people are having the maximum maths marks. So, it is not, if you have two people with maximum maths marks, then and since we kept only one variable which is maxCardId, therefore you cannot carry two card numbers there. So, we have to basically let maxCardId not only represent a single thing but probably a list of cards by separated let us say by comma or something like that. So, we can keep track of the not just the card that we found it but the list of cards on which we found the maximum number.

So, if do not do that, only the first max marks will be retained and the subsequent max, the max numbers that you encounter will simply be ignored. Because this condition is checking whether a card is greater than max, so if you find something that is equal to max that one will get ignored. So, you will go down the no path when you find a card like that. So if you want to keep track of the list of cards on which you found the maximum, not only should you extend this maxCardId to store a list but also you should check not only for greater but you have to check whether the current card which is card X, the maths marks is equal to max.

(Refer Slide Time: 10:22)

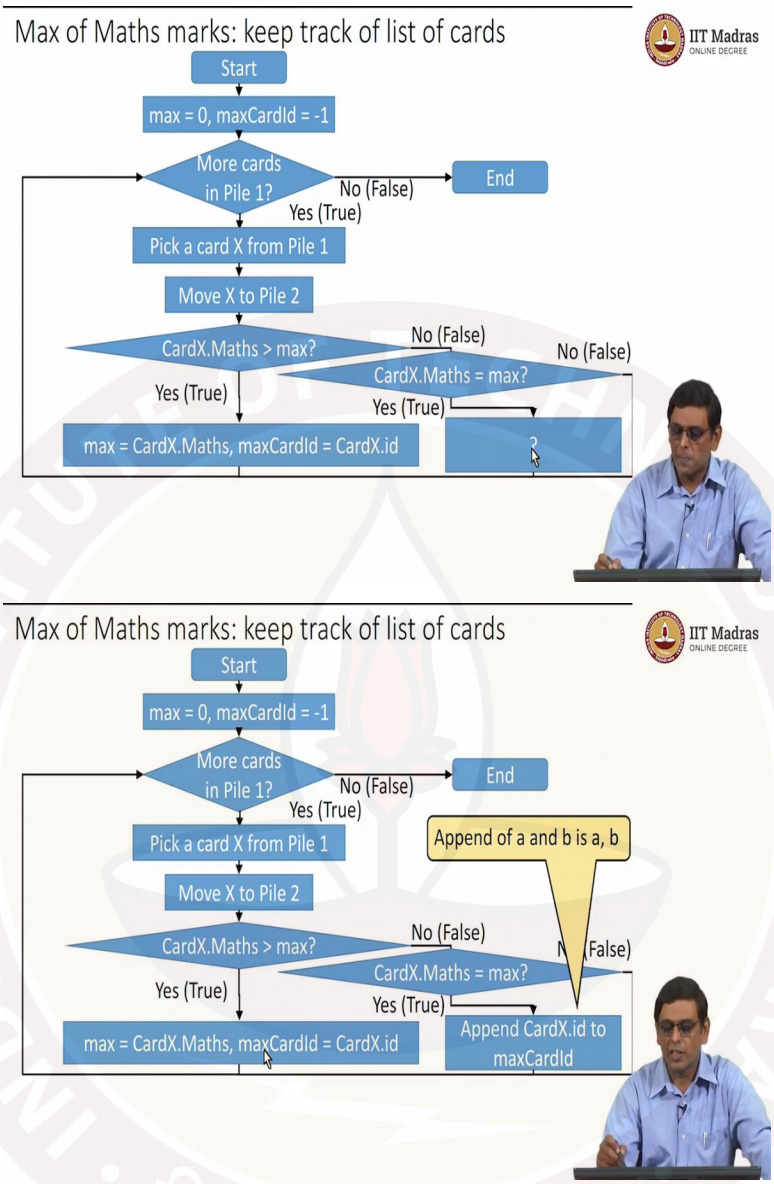




So, you will need one more check, and that is what I have done here. So, if you find that the card X is maths marks is not greater than max, you cannot just go back like this, you need to do one more check, which is to check if the card X the maths marks basically is equal to max. Is it right? So, in order to, so here I have used this symbol equal to basically to check whether the two things are equals, just like the greater than is used to check whether one is greater than the other and so on.

You could have other symbols that are used for checking equality, we will discuss this later just so that we do not confuse this update equal to here with the equal to equal here. But you know for now let us assume that this equal to is checking whether these two are equal or not. So, Cardx.Maths is equal to max, checking basically whether these two are equal. If they are, then we have to do something, if they are not then we do not need to do anything. So, we have to go basically like this, go (back), go like this.

(Refer Slide Time: 11:22)



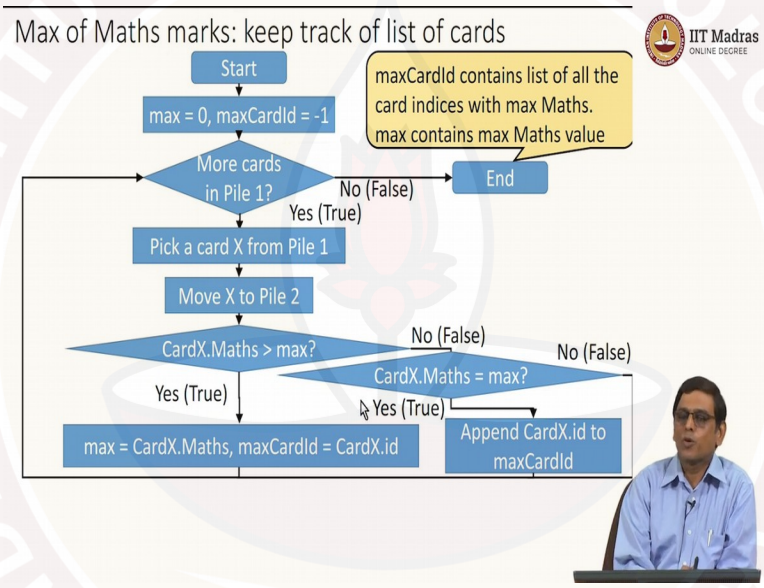
So, if it is not equal then you go back to the beginning of iteration. Nothing to be done. On the other hand, if it is true which means CardX.Maths is equal to max, then you need to do something in this box. And what you need to do in this box, you have a list which you are maintaining in maxCardId, you have to add to that list the Id of the card that you have seen, card X.

So, we are writing this thing called Append CardX.id to maxCardId, when Append basically will take two numbers a and b, and create a comma or to if a is already a list and b is a number it will create a comma b, which is a longer list it will create. On the other hand, if a is a number and b is

a number then it creates a comma b. So, it is basically creating a list, that is what append is doing. So, Append card X, so we are basically maxCardId as a list already and we are adding to that list this CardX.id, the new (card), that new Id that we have seen.

Now, of course, at this point if you find another, if you go around and find another card which has a larger max value, then you will overwrite, basically you will replace this list with the card, with the Id that you are seeing for the first time. So, when you see a new max for the first time, that max is stored in card Id and when you see further maxes, which are in the list then those maxes are appended. So, that is how you create a list of max card which in the CardX.id list.

(Refer Slide Time: 12:48)



So, when you finish the procedure, which is basically which happens when you come to this end terminal, at this time the max, max maths, the max variable will contain the maximum value of maths that you have seen. And maxCardId will contain the list of all the card indices which you have seen, which have, which has this max marks. So, which is what we need.

So, what we have seen through this particular lecture is procedure to take the flowchart for filtering, iteration with filtering. And modify that filtering procedure, so that we can compare it with the max variable, so that, that can then become some procedure to find the maximum. But also at the same time we are checking and doing the few other things to keep track of the list of cards on which that maximum value has been found.