




**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 – 3.8**  
**Pseudocode for three prizes problem**

So, now that we have seen the notation for procedures, let us try and write Pseudocode for one of the more complicated examples that we did in class, the one where we avoided awarded prizes to the top three students in the class.

(Refer Slide Time: 00:27)

**Awarding three prizes** 

**Want to award prizes to top 3 students**

- Basic criterion is total marks
- Must also be within top three in at least one subject
- Must select at least one boy and one girl for top three prizes

**Finding maximum**

- Initialize max to 0, scan cards, update each time you see a bigger value

**Finding top two marks**

- Maintain two values, max and secondmax

**Basic pattern**

- Find top three marks in a category
- How is this to be done?

*Handwritten notes: max, secondmax*

Pseudocode: Awarding three prizes

So, our goal was to award top three prizes and basically we wanted to give these prizes to the students with the maximum marks, but then we had some side conditions. So, one of these was that it is not enough to be in the top three overall, you must be in the top three in at least one of the three subjects.

So, unless you have done well in at least one subject and this could happen because we said there could be a lot of variation marks. People who do well in physics may not do well in chemistry and so on. So, we did not want somebody who had done averagely across all the things, but somehow got a higher total because the people who did well in one subject, did very poorly in the other subjects.

The other thing we said is we wanted some kind of equity across gender. So, we wanted to ensure that among the top three we select, there is at least one boy and at least one girl; we do not want all three boys, we do not want all three girls.

So, let us first look at this question about how to find the top three in a given category because that is common to both parts of this. So, there are two parts if you notice; one is to find the top three marks in each given subject because we need to check that the student is a subject wise topper and then we need the top three marks overall among the subject wise toppers.


So, we will repeatedly need to find the top three marks; so, let us see how to find this right. So, how do we find the top three marks in the category? Well, how to find the top mark in a category we know, this was the very first thing that we did; finding the maximum. So, we keep track of a variable which keeps track of the maximum by initializing it to 0 and whenever we see a new value, if it exceeds the current maximum; we update it.

So, maximum is something we have seen before, what about second maximum? Supposing, we want the top two; well, if you want the top two then we really need to keep track of two values because we need the top value and the second value and now the update is a little more complicated. So, in the maximum case we updated whenever we saw a new value which was bigger than the current maximum. So, only one situation, but now we have maximum and we have second maximum.

So, we have two values and what could happen is that the new value that we get could come here which is like updating the maximum or it could come here which is that the maximum is not touched, but the second maximum is now replaced by the new value. So, there are two updates that we have to consider in this case and we have to do them a little carefully.

(Refer Slide Time: 02:45)

### Awarding three prizes

IIT Madras  
ONLINE DEGREE

Want to award prizes to top 3 students

- Basic criterion is total marks
- Must also be within top three in at least one subject
- Must select at least one boy and one girl for top three prizes

Basic pattern

- Find top three marks in a category
- How is this to be done?


Finding maximum

- Initialize max to 0, scan cards, update each time you see a bigger value

Finding top two marks

- Maintain two values, max and secondmax
- If current card value is bigger than max
  - Copy max to secondmax, update max to current value
- If current card value is between max and secondmax
  - No change in max, update secondmax to current value

Pseudocode: Awarding three prizes



So, if the current value is bigger than the maximum; then we have to first before we overwrite the maximum, before we copy the value of the current card into the maximum; we have to remember that the maximum now becomes a second maximum.

So, we first copy the second maximum the maximum value to the second maximum variable and only then do we update the maximum. So, this is the first kind of update where the new card is bigger than anything we have seen before. And the other situation is when the new card comes in between, in this case the maximum does not change and we just replace a second maximum, as we would have done when we are keeping track of only one value.

(Refer Slide Time: 03:25)

Top three marks in a subject

- Maintain max, secondmax, thirdmax
- Scan through all the cards
- Update max, secondmax, thirdmax as appropriate

Procedure TopThreeMarks(Subj)



max = 0  
secondmax = 0  
thirdmax = 0  
while (Pile 1 has more cards) {  
    Pick a card X from Pile 1  
  
}  
End TopThreeMarks

max

secondmax

thirdmax

Pseudocode: Awarding three prizes



So, with this background, let us try to generalize it to finding the top three marks. So, as you would imagine you would keep track of three quantities; the maximum, the second maximum and the third maximum. So, let us directly try to write pseudocode for a procedure which finds the top three marks in a subject.

So, this takes the subject as a parameter and is going to compute the maximum, second maximum and third maximum marks in that subject across all the cards and this is what we will be using to find subject wise toppers as part of our three price problem.

So, the first step is to compute these; is to set up these three values and initialize them to 0. And now the usual thing happens which is we scan through all the cards and we do something. So, what is it that we do? Well, we have to update the maximum, the second maximum and third maximum depending on what the current card value has. So, like we saw for the second maximum, there will be multiple cases; now we have three values, we have max, we have second max and we have third max right.

So, now the interesting case is when either the new value is bigger than the maximum, in which case everything changes or it is in between these two in which case these two values change or it is here, in which case only this value changes. So, we have to do three separate types of updates depending on the current value; of course, if it is in the last position, if it is smaller than third maximum; then it is of no interest to us we just move on to the next card right.

(Refer Slide Time: 04:51)


Top three marks in a subject

- Maintain max, secondmax, thirdmax
- Scan through all the cards
- Update max, secondmax, thirdmax as appropriate
- Current value is
  - Bigger than max

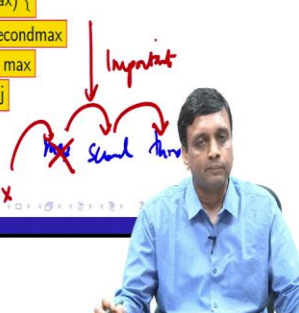
Procedure TopThreeMarks(Subj)

```
max = 0
secondmax = 0
thirdmax = 0
while (Pile 1 has more cards) {
  Pick a card X from Pile 1
  if (X.Subj > max) {
    thirdmax = secondmax
    secondmax = max
    max = X.Subj
  }
}
```

End TopThreeMarks



IIT Madras  
ONLINE DEGREE



Handwritten notes: 'Important' with an arrow pointing to the update logic, and 'Scan through' with an arrow pointing to the while loop. A person is visible in the bottom right corner of the slide.

So, the first situation is when the subject of the subject field. So, remember the subject is an argument; so it could be maths, it could be chemistry, it could be physics; when the subject that we are looking for is bigger than the maximum, then we have to update all three. And now the order is important we have to first; so we have max, second and third right. So, we have to first make sure that we copy this value there and then copy this value here and then copy the new card value here.

Because, if we do it in another order; if we first do this for instance, then max; the old value of max is gone, then we cannot copy it to second anymore; so this order is important right.

We have to start with the value that you are willing to discard, in this case the third max is no longer useful to us; we are willing to discard it. So, we overwrite third max with the second max, then we overwrite the second max with the max and then we finally, replace the max with the current card value because the current card value by assumption is bigger than the max.




(Refer Slide Time: 05:49)

Top three marks in a subject


- Maintain max, secondmax, thirdmax
- Scan through all the cards
- Update max, secondmax, thirdmax as appropriate
- Current value is
  - Bigger than max
  - Between max and secondmax

Procedure TopThreeMarks(Subj)

max = 0  
secondmax = 0  
thirdmax = 0  
while (Pile 1 has more cards) {  
    Pick a card **X** from Pile 1  
    ...  
    if (max > **X.Subj** > secondmax) {  
        thirdmax = secondmax  
        secondmax = **X.Subj**  
    }  
}  
End TopThreeMarks



IIT Madras  
ONLINE DEGREE



Pseudocode: Awarding three prizes

The second situation is when it falls in the first interval between max and second max. So, again we do two updates; we update the third max with the second max, we throw away the third max. We replace the second max with the current value and the overall max does not change.


(Refer Slide Time: 06:05)

Top three marks in a subject


- Maintain max, secondmax, thirdmax
- Scan through all the cards
- Update max, secondmax, thirdmax as appropriate
- Current value is
  - Bigger than max
  - Between max and secondmax
  - Between secondmax and thirdmax

Procedure TopThreeMarks(Subj)

max = 0  
secondmax = 0  
thirdmax = 0  
while (Pile 1 has more cards) {  
    Pick a card **X** from Pile 1  
    ...  
    if (secondmax > **X.Subj** > thirdmax) {  
        thirdmax = **X.Subj**  
    }  
}  
End TopThreeMarks



IIT Madras  
ONLINE DEGREE




Pseudocode: Awarding three prizes

And finally, if it is in the third interval that is between the second max and the third max, then only the third max has to be updated. So, these are the three different types of

updates that occur because we are looking for the top three marks. And the fourth case when it is smaller than third marks is where we just ignore the card and move ahead.

(Refer Slide Time: 06:21)

Top three marks in a subject





- Maintain max, secondmax, thirdmax
- Scan through all the cards
- Update max, secondmax, thirdmax as appropriate
- Current value is
  - Bigger than max
  - Between max and secondmax
  - Between secondmax and thirdmax
- Need to return three values as a list  $[v_1, v_2, v_3]$ ? Lists later ...

```

Procedure TopThreeMarks(Subj)
    max = 0
    secondmax = 0
    thirdmax = 0
    while (Pile 1 has more cards) {
        Pick a card X from Pile 1
        ...
    }
    return([max, secondmax, thirdmax])
End TopThreeMarks

```





So, at the end of this procedure; this loop rather, what we have is three values; we have the max, second max and the third max and remember that now this is a procedure it has to return these values back to us so, we can use them.

So, this is slightly complicated compared to what we had before because we not; we do not have just one value to return to us, we have three values right; we have three numbers and how do we return three numbers and how do we store them? So, technically for this we need to have some way of collecting these three numbers into a sequence or a list and passing it back and this will come up later in the course.

So, we will not look at this now because this requires some new notation and some complications which is best to avoid. So, we will see list later on, but fortunately for the reason, for the purpose that we are using this procedure; we actually do not need all three values. What do we want to check right? We want to check that the student card that we are looking at, is a topper meaning it is at least as good as the third highest mark.

So, we really have no interest in the maximum and the second maximum; we want to check whether the student is at least as good as the third maximum. So, the third maximum is the only interesting value in this, as far as this three price problem is



concerned. We have no interest in whether the person is a topper or not in that particular category; we just need to know that they are within the top three.

(Refer Slide Time: 07:41)

Top three marks in a subject

- Maintain max, secondmax, thirdmax
- Scan through all the cards
- Update max, secondmax, thirdmax as appropriate
- Current value is
  - Bigger than max
  - Between max and secondmax
  - Between secondmax and thirdmax
- Need to return three values as a list  $[v_1, v_2, v_3]$ ? Lists later ...
- Sufficient to return thirdmax

Procedure **TopThreeMarks(Subj)**


```
max = 0
secondmax = 0
thirdmax = 0
while (Pile 1 has more cards) {
    Pick a card X from Pile 1
    ...
}
return(thirdmax)
End TopThreeMarks
```

Pseudocode: Awarding three prizes

So, fortunately for us; we can replace this list by just a single value the third max. Because once we know the third max, we can discover whether the person is a subject topper without having to go through this other information. So, with this procedure, we now know how to compute the top three marks and in particular return the third marks. But, since we did it piecemeal; it is a good idea to probably see the procedure as a whole so that you see exactly how the whole thing is laid out.


(Refer Slide Time: 08:07)

Top three marks in a subject, in entirety



```
Procedure TopThreeMarks(Subj)
    max = 0
    secondmax = 0
    thirdmax = 0
    while (Pile 1 has more cards) {
        Pick a card X from Pile 1
        if (X.Subj > max) {
            thirdmax = secondmax
            secondmax = max
            max = X.Subj
        }
        else if (max > X.Subj > secondmax) {
            thirdmax = secondmax
            secondmax = X.Subj
        }
        else if (secondmax > X.Subj > thirdmax) {
            thirdmax = X.Subj
        }
    }
    return(thirdmax)
End TopThreeMarks
```

Pseudocode: Awarding three prizes



So, this is the entire procedure to compute the top three marks in a subject. So, in the beginning; we did this initialization, we took these three values and initialized them to 0. And now this overflows into two columns, so think of it as two pages; so we have a while loop which goes from here to here right. So, this while loop iterates over all the cards in our deck, so it goes card by card until no cards are left.


So, we pick up each card and now there are three different types of updates which are disjoint right. Either, it is bigger than the max in which case you update all three values or it is between max and second max in which case you update two values or it is between second max and third max, in which you update only one value. And in the fourth case where its smaller than the third max, you just ignore this card and move ahead.

And finally, at the end of this procedure, you return the third max value that you found because this is what is important for a; in order to decide whether somebody is a subject topper or not.

(Refer Slide Time: 09:05)


### Three prizes


- Top three totals such that top three in at least one subject
  - Deal with boy/girl requirement later
- Again, maintain and update max, secondmax, thirdmax
- Scan through all the cards
- For each card, update max, secondmax, thirdmax as before
  - But only if in the top three of at least one subject!
  - Record third highest mark in each subject
  - Compare with subject marks before updating max, secondmax, thirdmax
- After scanning all cards, we have three prize winning totals
  - But who are the winners?
  - Keep track of card number of prize winners



IIT Madras  
ONLINE DEGREE

Pseudocode: Awarding three prizes





So, now we come back to the three prize problem; so, we want to find the top three totals such that these totals also include a mark which is in the top three in at least one subject. We also have a gender criterion, but the gender criterion is hard to maintain because until we find the overall top three marks, we really do not know who is going to be the final three; so, we cannot really keep track of the boys and the girls in this list, until we compute the top three.

So, we will postpone this issue about boys and girls till the end when we compute the top three and if it is not correct, we will figure out how to fix it. So, right now what we want to do is find the top three totals which are also subject toppers. So, once again we will maintain an update; max, second max and third max and scan through all the cards. And for each card, we will update this max, second max and third max provided that particular card is a subject topper. If it is not a subject topper, even it has; if it has a total which is high, we will ignore it.

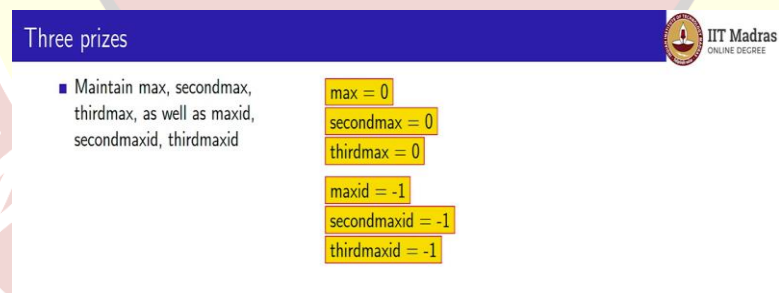
So, in order to compute this subject topper criterion; we need to first find out the subject toppers and save them because we do not want each time to go through and find the top three marks in each subject, for each card that we look at. So, what we will assume is that we will call our previous procedure which I am finds the top three marks, we will call it one for maths, one for physics, one for chemistry and we will store the third highest mark in each of these subjects.

Then, when we look at a new card; we can compare it against this third highest marks and quickly decide whether or not this particular card qualifies to be a prize winner. And if it qualifies to be a prize winner, then we will check whether we want to update max, second max and third max for the total marks. So, if we do this which we will see in detail; at the end we will have the three prize winning totals, just like as before we had the three highest marks in each subject.

But the situation here is different; it is not really enough for us to have the three prize winning totals because we need to know who the winners are right. Finally, it is not good enough to know that the top mark was 281 on 300, we need to know; who had 281 on 300. So, just like we saw when we did the maximum right at the beginning, there is a problem of finding the maximum marks in the subject, but there is also a related problem for identifying who it is?

So, we need to keep track of the id of the card which contributed this maximum. So, we need to keep track of the card number of the prize winners, along with the marks of the prize winners as we go along. So, that is another difference between the three prize version of finding the top three and finding the top three in a given subject.

(Refer Slide Time: 11:39)



Three prizes

■ Maintain max, secondmax, thirdmax, as well as maxid, secondmaxid, thirdmaxid

```
max = 0
secondmax = 0
thirdmax = 0
maxid = -1
secondmaxid = -1
thirdmaxid = -1
```

IIT Madras  
ONLINE DEGREE



Pseudocode: Awarding three prizes

Video feed of a lecturer in a blue shirt.

So, let us look at the pseudocode now, this is not going to be a separate procedure; this is our main code; so we will just write it as a sequence of steps where we will of course,

use the procedures that we had. So, the first thing is to set up our three variables for the marks; max, second max and third max.

But we also have three variables for the id's; max id, second max id, third max id because we need to keep track of which cards contributed these values. And the id's we are assigning to minus 1, to make sure that they are not any of the card id's which are already in the deck; so, we are pretty sure that the cards are not; do not have negative numbers on them as their id's. So, we assign them to minus 1, so we are not confused whether this is a valid id or not; if we see a minus 1, we know that there is no card which is seen so far.

(Refer Slide Time: 12:23)

Three prizes

- Maintain max, secondmax, thirdmax, as well as maxid, secondmaxid, thirdmaxid
- Record third highest mark in each subject

< Initialization of max, maxid etc >

```
maths3 = TopThreeMarks(Maths)
phys3 = TopThreeMarks(Physics)
chem3 = TopThreeMarks(Chemistry)
```

Pseudocode: Awarding three prizes

IIT Madras  
ONLINE DEGREE

So, that is our first step; we have initialized these six variables. Now, what we said was that we do not want to recompute these third highest marks in each subject again and again. So, we do it once and for all in the beginning and we remember these values in three variables.

So, we have three variables maths 3, physics 3, chemistry 3; each of which records the third highest marks in the appropriate subject, by calling our previous procedure top three marks with the appropriate subject name. So, this is part of our setup; so the first setup was just initializing the variables we will be using to keep track of the top three prize winning totals. And then the second setup is to keep track of the marks that we

need in order to decide whether a given card qualifies or not under the subject topper criteria.

(Refer Slide Time: 13:07)

**Three prizes**

- Maintain max, secondmax, thirdmax, as well as maxid, secondmaxid, thirdmaxid
- Record third highest mark in each subject
- Scan through all the cards
- Update max, secondmax, thirdmax as appropriate
- Only if top three in some subject — new procedure  
`SubjectTopper(...)` → True / False

*(Initialization of max, maxid etc )  
( Record third highest per subject )*

```
while (Pile 1 has more cards) {  
  Pick a card X from Pile 1  
  if (SubjectTopper(X, math3, phys3, chem3)) {  
    ...  
    ...  
  }  
}
```

Pseudocode: Awarding three prizes

IIT Madras ONLINE DEGREE

So, now we have our familiar loop where we pass through all the cards and we do something and that something is to update the max, second max and third max for the totals, but remember that we do this only if the card qualifies as a subject topper.


So, we have to check whether the marks on this card, for each of these three subjects; whether at least one of them is above the third highest. And since we do not want this to be computed again and again, we assume that we will use the values we have already stored in these math3, physics3 and chem3 variables.

So, for the moment to make it simpler; we will assume that we have a procedure which we will call with the current card and with these three thresholds, these three third highest marks. And it will return back to us either a true saying yes, this card is a subject topper or false saying; no, this card is not a subject topper. If it is not a subject topper, we have nothing to do; we can just ignore it and go ahead, even if it has a high total.



(Refer Slide Time: 14:13)

### Three prizes



■ Maintain max, secondmax, thirdmax, as well as maxid, secondmaxid, thirdmaxid

■ Record third highest mark in each subject

■ Scan through all the cards


■ Update max, secondmax, thirdmax as appropriate


- Only if top three in some subject — new procedure  
**SubjectTopper(...)**

( Initialization of max, maxid etc )  
( Record third highest per subject )

```
while (Pile 1 has more cards) {  
    Pick a card X from Pile 1  
    if (SubjectTopper(X, math3, phys3, chem3)){  
        if (X.Total > max) {  
            thirdmax = secondmax  
            thirdmaxid = secondmaxid  
            secondmax = max  
            secondmaxid = maxid  
            max = X.Total  
            maxid = X.Id  
        }  
    }  
}
```

Pseudocode: Awarding three prizes






If it is a subject topper, now we have to do this multi way update depending on the total marks of this card. So, the first case as before is when the total is bigger than the overall maximum. Then, we have to do this three step thing; we first replace the third maximum values, then we replace the second maximum values and then we replace the maximum values, but now the difference is that we also have to do this for the id's right.

So, earlier we are only dealing with the totals, now we are also replacing the id's. So, it is a slightly more involved update; each time, we have to update the marks and the id in each category.

(Refer Slide Time: 14:43)


### Three prizes



- Maintain max, secondmax, thirdmax, as well as maxid, secondmaxid, thirdmaxid
- Record third highest mark in each subject
- Scan through all the cards
- Update max, secondmax, thirdmax as appropriate
  - Only if top three in some subject — new procedure **SubjectTopper(...)**

```
{ Initialization of max, maxid etc }
{ Record third highest per subject }
while (Pile 1 has more cards) {
  Pick a card X from Pile 1
  if (SubjectTopper(X, math3, phys3, chem3)){
    ...
    if (max > X.Total > secondmax) {
      thirdmax = secondmax
      thirdmaxid = secondmaxid
      secondmax = X.Total
      secondmaxid = X.Id
    }
  }
}
```


Pseudocode: Awarding three prizes



The second situation is when it lies between the second; first and the second max, in which case we update only two quantities; the third max and the second max.

(Refer Slide Time: 14:53)


### Three prizes



- Maintain max, secondmax, thirdmax, as well as maxid, secondmaxid, thirdmaxid
- Record third highest mark in each subject
- Scan through all the cards
- Update max, secondmax, thirdmax as appropriate
  - Only if top three in some subject — new procedure **SubjectTopper(...)**

```
{ Initialization of max, maxid etc }
{ Record third highest per subject }
while (Pile 1 has more cards) {
  Pick a card X from Pile 1
  if (SubjectTopper(X, math3, phys3, chem3)){
    ...
    if (secondmax > X.Total > thirdmax) {
      thirdmax = X.Total
      thirdmaxid = X.Id
    }
  }
}
```


Pseudocode: Awarding three prizes



And the final situation is when it lies below the second max, but above the third max; in which case we just replace the third max.

(Refer Slide Time: 15:01)

### Three prizes

IIT Madras  
ONLINE DEGREE

- Maintain max, secondmax, thirdmax, as well as maxid, secondmaxid, thirdmaxid
- Record third highest mark in each subject
- Scan through all the cards
- Update max, secondmax, thirdmax as appropriate
  - Only if top three in some subject — new procedure
- In the end, we have what we need

```
{ Initialization of max, maxid etc }  
{ Record third highest per subject }  
while (Pile 1 has more cards) {  
  Pick a card X from Pile 1  
  { Update max, maxid etc }  
}
```

Variables of interest

- maxid, max
- secondmaxid, secondmax
- thirdmaxid, thirdmax

SubjectTopper(...)

Pseudocode: Awarding three prizes




So, now if we go through this; at the end of this, we have the six variables that we kept track of. For each variable, we had the marks, we had the max marks and the max id; we had the second max mark and second max id, third max marks and the third max id.

Now, actually notice that if we have the id's, we can always go back to the card and get the marks. So, we do not really technically speaking need the marks because we can go back, but then we have to go back right. So, we have to go and look up the card again; in order to get the marks belonging to that id.

So, though it is enough to keep track of the id's; it is useful to keep track of the max, marks and the id for each of these categories. So, we want to finish off this code, but we have this hanging subject topper. So, let us see what subject topper looks like.

(Refer Slide Time: 15:47)


Subject topper

- Compare each subject's marks on card with third highest
  - Passed explicitly as parameters
- One or more comparisons should succeed — **or** operator
- Value returned is a Boolean — True or False
- Typically, we would call this as follows:  
  
`if (SubjectTopper(X,M,C,P) == True)`

Procedure SubjectTopper  
(Card,MMark,PMark,CMark)

```
if (Card.Maths ≥ MMark or
    Card.Physics ≥ PMark or
    Card.Chemistry ≥ CMark) {
    return(True)
}
else {
    return(False)
}
End SubjectTopper
```

Pseudocode: Awarding three prizes



So, subject topper requires us to compare each subjects marks with the third highest. So, we have a card right and then we have these three values, so we what we want to check is whether the maths max on the card is bigger than the third highest maths mark or the third highest physics mark, for the physics value or the third highest chemistry mark for the chemistry value.

Now, notice that this particular procedure; subject topper does not really know that these are the third highest marks. So, for instance, tomorrow you could also use it if you wanted the fifth highest marks; all you have to do is pass the right parameters. All that the subject topper procedure is doing is checking that the maths marks are above the math threshold. The physics marks above the physics threshold and the chemistry marks is above the chemistry threshold.

What their threshold is; this procedure does not care, it is just telling you that it at least one of these is above the given threshold in that subject. So, since we want this at least right; we need to compare these and this is done using this or operator. So, this or operator basically tells us that one of these condition should hold, more than one could hold, but at least one should hold.

So, either it should be that the match marks is above the threshold or the physics marks or the chemistry marks. Now, what is this; this function return? As we said, if the threshold is true for at least one subject, it will return true; otherwise it will return false.

So, when we go back to the function or the procedure that called this, technically what it would have to do is; it would have to call this procedure and check the value against true or false.

So, we would have to say is subject topper of a given card X with respect to the maths, chemistry and physics marks is it true, but the outcome of this is again going to be true or false right. So, when is it going to be true? It is going to be true, when the outcome of the subject is true; when is it going to be false? When, the outcome of subject topper is false. So, if subject topper returns false, this equality check will return false; if subject topper returns true, then this equality check will return true.

So, we actually do not need the equality check; we can just take directly the value that subject topper gives us, if it is false; this test is false, if it is true; the test is true and use that as our condition, that is what we wrote in our code.

(Refer Slide Time: 17:57)

**Subject topper**

- Compare each subject's marks on card with third highest
  - Passed explicitly as parameters
- One or more comparisons should succeed — **or** operator
- Value returned is a Boolean — True or False
- Typically, we would call this as follows:  
`if (SubjectTopper(X,M,C,P) == True)`
- Return value is Boolean, so ...  
`if (SubjectTopper(Id,M,C,P))`

**Procedure SubjectTopper**  
**(Card,MMark,PMark,CMARK)**

```
if (Card.Maths ≥ MMark or
    Card.Physics ≥ PMark or
    Card.Chemistry ≥ CMARK) {
    return(True)
}
else {
    return(False)
}
End SubjectTopper
```


Pseudocode: Awarding three prizes

So, we can; we can eliminate this equal to equal to true because it is only the value returned by this function which is important to this procedure right. This procedure tells us true or false, if it is true; we do something, if it is false; we do not do something. So, this is a small simplification of the pseudocode; it also makes it easier to read. So, now we have the subject topper pseudocode; so we are now ready to write out the pseudocode for the three prize problem, in more detail right.




(Refer Slide Time: 18:23)

Three prizes, in entirety



```
max = 0
secondmax = 0
thirdmax = 0
maxid = -1
secondmaxid = -1
thirdmaxid = -1
maths3 = TopThreeMarks(Maths)
phys3 = TopThreeMarks(Physics)
chem3 = TopThreeMarks(Chemistry)
while (Pile 1 has more cards) {
    Pick a card X from Pile 1
    if (SubjectTopper(X, maths3, phys3, chem3)) {
        if (X.Total > max) {
            thirdmax = secondmax
            thirdmaxid = secondmaxid
            secondmax = max
            secondmaxid = maxid
        }
        else if (max > X.Total > secondmax) {
            thirdmax = secondmax
            thirdmaxid = secondmaxid
            secondmax = X.Total
            secondmaxid = X.Id
        }
        else if (secondmax > X.Total > thirdmax) {
            thirdmax = X.Total
            thirdmaxid = X.Id
        }
    }
}
```

Pseudocode: Awarding three prizes



So, here is the full three prize problem; so we have these two initializations; the max values and the max id's. Then, we have this three calls to the top three marks procedure that we had before to up set up the three thresholds and then we have our familiar loop right.

So, we go through all the cards up to here and for each card; we first check whether or not it is a subject topper. If it is not a subject topper and this; this ends here, if it is not a subject topper, we just skip the card; we do nothing, we do not look at the total at all. If it is a subject topper, then we scan the total in three ways as before, we check whether the total is bigger than the max or we check whether total is between max and second max or we check whether the total is between second max and third max.

And at the end of this we have these three values; max, second max, third max and the corresponding id's; max id, second max id and third max id.



(Refer Slide Time: 19:23)

#### Boundary conditions



- What if all prize winners are of the same gender?
- Exclude the third prize winner and repeat the process
  - How many times?
  - Till we get three prize winners with at least one boy and one girls
  - Will this always given, us three valid prize winners?
- What if there are ties?
  - How many ties can we tolerate?
  - Does it depend on first, second or third position?



Pseudocode: Awarding three prizes

So, now let us come up to this question about gender. So, we have computed these top three marks; what if they are not of different genders? Well, we have to decide what we want to do, but the best solution; obviously, is to knock off the smallest of these three; right, to take off the. So, supposing they are all three boys; then we take off the third boy because we know that the top girl is below the third boy.

So, the top girl is at most fourth overall; so, then we go through and we look again having eliminated the third most; third highest boy, in the hope that now this third highest boy will be replaced by a girl. So, how many times we do it? Well, we have to keep doing it until we find a card with a different gender, copying up in the top three. So, this is something which we have to check now; will we always give get, will this always give us three valid prize winners right?

Now, of course, if all the girls or among the girls, there is a somebody who has; he is a subject topper, you can check that you will get it or what if none of the girls is a subject topper? If none of the girls is a subject topper, then you will never find a girl who qualifies for the prize because that is the first criterion that unless you are a subject topper, your total is not even counted.

So, there would be a situation in which we keep doing this and finally, we find that there is no way to award a prize. So, we have to then go back and revisit maybe how we have chosen these prize criteria. So, these are some issues that come up when you formulate a

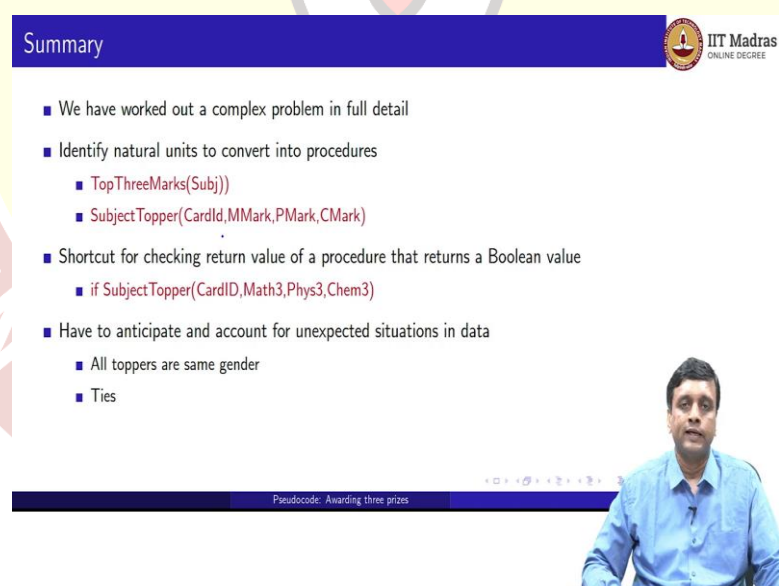
problem. We initially come up with some criteria and then you later on realize that these criteria need not be sound.


The other thing that we have to look at is ties; see what if there are marks two students who have the same number of marks? So, if there are ties for the top place for instance; if there are three people who are tied for the top place, they will get the first three prizes. What if there are four people? We have a problem.

Similarly, if there are two people tied for the second place; it is ok, we will give one person for the first place and the second and third will get; the two people for tied for second place will get two prizes, but what if there are three? So, therefore, the number of ties we can tolerate in this procedure also depends on whether the ties happen at the first position, the second position or the third position.

So, once again if you have a lot of ties; then this procedure will result in some arbitrary choice about who gets the prize and who does not get the prize.


(Refer Slide Time: 21:55)



**Summary** 

- We have worked out a complex problem in full detail
- Identify natural units to convert into procedures
  - `TopThreeMarks(Subj))`
  - `SubjectTopper(CardId,MMark,PMark,CMARK)`
- Shortcut for checking return value of a procedure that returns a Boolean value
  - `if SubjectTopper(CardID,Math3,Phys3,Chem3)`
- Have to anticipate and account for unexpected situations in data
  - All toppers are same gender
  - Ties

Pseudocode: Awarding three prizes



So, to summarize; this lecture was an illustration of a reasonably complicated problem; worked out in detail in pseudocode. One of the steps that you must take away from this is that; it is very useful to break up your pseudocode or your procedure into natural units right. So, you want to take the main procedure and block it off into repeated things; like

we found here, the top three marks again and again or we found the subject topper by not having to call it again and again.

So, make these into separate procedures; so it makes it easier to modularize and read your code. We also saw the shortcut for looking at values returned by procedures which give back Boolean values. And finally, we saw that in the situation of gender or with ties; we have to sometimes look back and see whether the problem that we have solved has been solved correctly and in some sense whether the solution can work in all cases or not.

