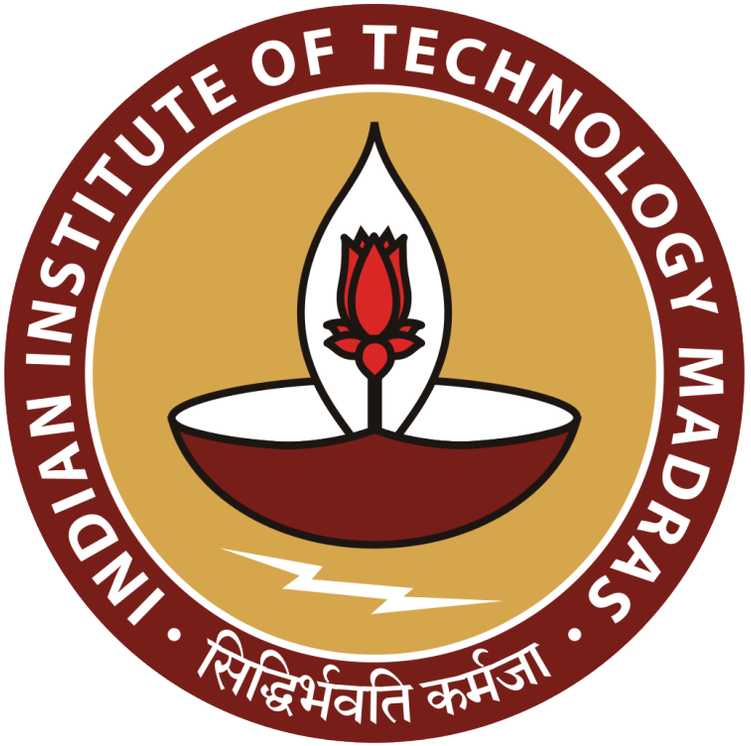

IIT Madras

ONLINE DEGREE

Pseudocode: Awarding three prizes

Awarding three prizes

Want to award prizes to top 3 students

- Basic criterion is total marks

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

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

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

Basic pattern

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

Basic pattern

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

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

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

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

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

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

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

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

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

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

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

Top three marks in a subject

- Maintain max, secondmax, thirdmax

Procedure TopThreeMarks(Subj)

max = 0

secondmax = 0

thirdmax = 0

End TopThreeMarks

Top three marks in a subject

- Maintain max, secondmax, thirdmax
- Scan through all the cards

Procedure TopThreeMarks(Subj)

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

End TopThreeMarks

Top three marks in a subject

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

Procedure TopThreeMarks(Subj)

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

End TopThreeMarks

Top three marks in a subject

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

Top three marks in a subject

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

Top three marks in a subject

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

Top three marks in a subject

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

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

Top three marks in a subject

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

Top three marks in a subject

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

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
      }
      if (max > X.Subj > secondmax) {
        thirdmax = secondmax
        secondmax = X.Subj
      }
      if (secondmax > X.Subj >
                          thirdmax) {
        thirdmax = X.subj
      }
    }
    return(thirdmax)
End TopThreeMarks
```

Three prizes

- Top three totals such that top three in at least one subject
  - Deal with boy/girl requirement later

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

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

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

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!

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

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

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?

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

Three prizes

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

max = 0
secondmax = 0
thirdmax = 0

maxid = -1
secondmaxid = -1
thirdmaxid = -1

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)

Three prizes

- Maintain max, secondmax, thirdmax, as well as maxid, secondmaxid, thirdmaxid
- Record third highest mark in each subject
- Scan through all the cards

⟨ Initialization of max, maxid etc ⟩
⟨ Record third highest per subject ⟩

```
while (Pile 1 has more cards) {
    Pick a card X from Pile 1
    ...
    ...
}
```

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

⟨ Initialization of max, maxid etc ⟩
⟨ Record third highest per subject ⟩

while (Pile 1 has more cards) {
  Pick a card X from Pile 1

  ...

  ...

}

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)){
    ...
    ...
  }
}
```

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
    }
  }
```

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
    }
  }
}
```

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
    }
  }
}
```

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(...)**
- 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

Subject topper

- Compare each subject's marks on card with third highest

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

Subject topper

- Compare each subject's marks on card with third highest
  - Passed explicitly as parameters

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

Subject topper

- Compare each subject's marks on card with third highest
  - Passed explicitly as parameters
- One or more comparisons should succeed — **or** operator

```
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
```

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

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

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

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(X,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

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,math3,phys3,chem3)){
    if (X.Total > max) {
      thirdmax = secondmax
      thirdmaxid = secondmaxid
      secondmax = max
      secondmaxid = maxid
      max = X.Total
      maxid = X.Id
    }
    if (max > X.Total > secondmax) {
      thirdmax = secondmax
      thirdmaxid = secondmaxid
      secondmax = X.Total
      secondmaxid = X.Id
    }
    if (secondmax > X.Total > thirdmax) {
      thirdmax = X.Total
      thirdmaxid = X.Id
    }
  }
}
```

Boundary conditions

- What if all prize winners are of the same gender?

Boundary conditions

- What if all prize winners are of the same gender?
- Exclude the third prize winner and repeat the process

Boundary conditions

- What if all prize winners are of the same gender?
- Exclude the third prize winner and repeat the process
  - How many times?

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

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?

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?

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?

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?

Summary

- We have worked out a complex problem in full detail

Summary

- We have worked out a complex problem in full detail
- Identify natural units to convert into procedures

Summary

- We have worked out a complex problem in full detail
- Identify natural units to convert into procedures
  - TopThreeMarks(Subj)

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)

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

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))

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

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

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