Lecture 4 - Conditionals

Alexander Lam

Conditionals

- if cond then A
  - Check the condition cond.
  - If it is true, then execute A.
  - Otherwise, do nothing.

```
if a < 0 then
    a = -a
```

- if cond then A else B
  - Check the condition cond.
  - If it is true, then execute A.
  - Otherwise, execute B.

```
if a < b then
    answer = a
else
    answer = b
```

Question 1

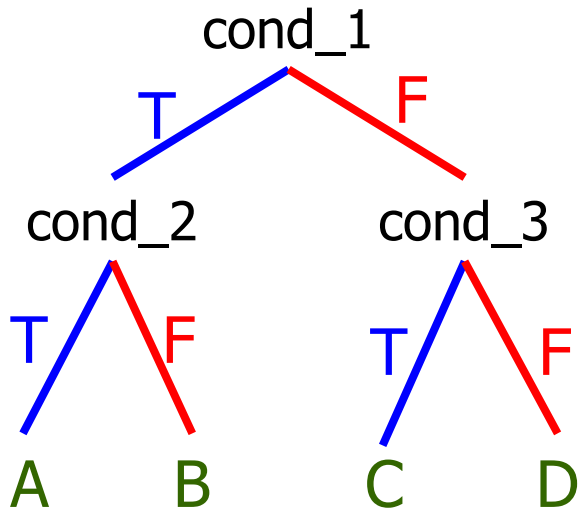
What does the following code print?

```
a = 50
if a > 50 then
    print(“ABC”)
else
    print(“123”)
if a < 50 then
    print(“DEF”)
else
    print(“456”)
```

Nested Conditionals

- Conditionals can be nested.

```
if cond_1 then
    if cond_2 then A
    else B
else
    if cond_3 then C
    else D
```



- Four types of situations.
  - Cond_1 true and cond_2 true $\Rightarrow$ A
  - Cond_1 true and cond_2 false $\Rightarrow$ B
  - Cond_1 false and cond_3 true $\Rightarrow$ C
  - Cond_1 false and cond_3 false $\Rightarrow$ D

Question 2

What is the final value of totalreps after the following code is run?

```
totalreps = 0
for weight in [130, 150, 170, 160, 140]
    reps = 0
    while alex is not tired
        reps = reps + 1
        alex = alexTired(reps, weight)
    totalreps = totalreps + reps
```

```
Function alexTired(reps, weight)
    if weight >= 170 then
        if reps >= 1
            return tired
        else return not tired
    else if weight >= 150 then
        if reps >= 3
            return tired
        else return not tired
    else
        if reps >= 5
            return tired
        else return not tired
```

Question 3

Given three unequal numbers, a, b, c, we want to print the correct ordering. Is the following code corr

```
if a > b then
    if b > c then
        print “a > b > c”
    else
        print “a > c > b”
else
    if b < c then
        print “a < b < c”
    else
        print “a < c < b”
```

```
if a > b then
    if b > c then
        print “a > b > c”
    else
        if a > c then
            print “a > c > b”
        else
            print “c > a > b”
else
    if b < c then
        print “c > b > a”
    else
        if a < c then
            print “b > c > a”
        else
            print “b > a > c”
```

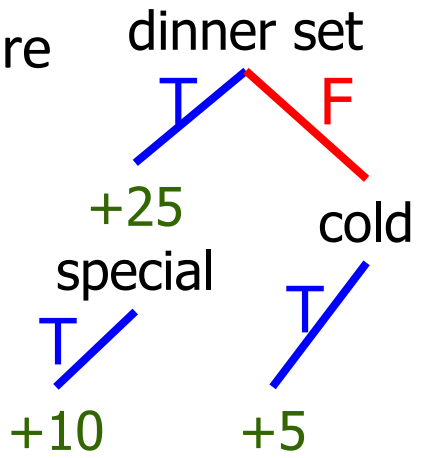
Indentation

- Consider ordering a dinner dish at a typical cha chaan teng.
  - Cost of dinner main dish is $80
  - Order dinner set with $25 more (salad + drink)
  - Special drink in dinner set can be ordered for $10 more (e.g. lemon coke or red bean ice)
  - All special drinks are cold drinks
  - Standard drink means coffee or tea
  - Standard cold drink at $5 more

Question 5

- Consider ordering a dinner dish at a typical cha chaan teng.
  - Cost of dinner main dish is $80
  - Order dinner set with $25 more (salad + drink)
  - Special drink in dinner set can be ordered for $10 more (e.g. lemon coke or red bean ice)
  - All special drinks are cold drinks
  - Standard drink means coffee or tea
  - Standard cold drink at $5 more

Is this pseudocode correct?




```
set cost to 80
if dinner set then
   cost = cost + 25
      if special drink then
         cost = cost + 10
else
   if cold drink then
      cost = cost + 5
return cost
```

Question 6

- Consider ordering a dinner dish at a typical cha chaan teng.
  - Cost of dinner main dish is $80
  - Order dinner set with $25 more (salad + drink)
  - Special drink in dinner set can be ordered for $10 more (e.g. lemon coke or red bean ice)
  - All special drinks are cold drinks
  - Standard drink means coffee or tea
  - Standard cold drink at $5 more

Write correct pseudocode

Old pseudocode

```
set cost to 80
if dinner set then
   cost = cost + 25
      if special drink then
         cost = cost + 10
else
   if cold drink then
      cost = cost + 5
return cost
```

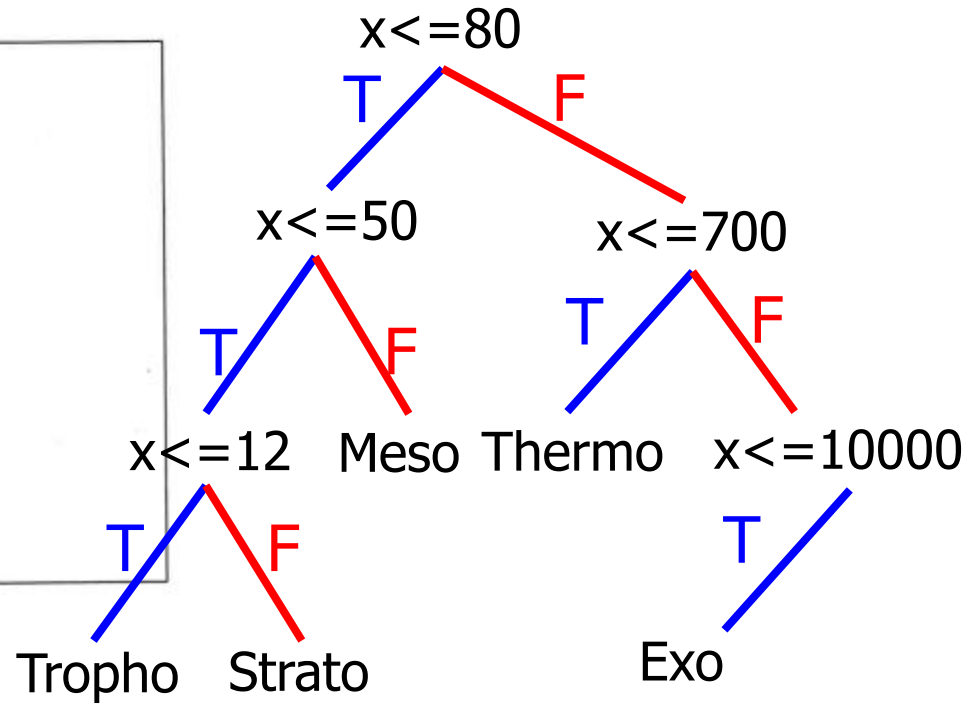
```
set cost to 80
if dinner set then
   cost = cost + 25
      if special drink then
         cost = cost + 10
      else
         if cold drink then
            cost = cost + 5
return cost
```

Question 7 (not on the Blackboard Quiz)

Draw a decision tree to illustrate the following pseudocode.

7. Examine the following pseudocode for determining the location of an object in the major layers of the Earth's atmosphere, based on its distance, x, in terms of kilometers, from the Earth's surface.

```
if x <= 80
      if x <= 50
            if x <= 12
                  return "Troposphere"
            else
                  return "Stratosphere"
      else
            return "Mesosphere"
else
      if x <= 700
            return "Thermosphere"
      else if x <= 10000
            return "Exosphere"
```



From 2022 past paper

Conditionals

- List of conditions
  - Count the number of times for each face of a die.

```
reset count1 to count6 value to zero
for i in [1..1000] do
   face = throw a die
   if face = 1 then count1 = count1 + 1
   if face = 2 then count2 = count2 + 1
   if face = 3 then count3 = count3 + 1
   if face = 4 then count4 = count4 + 1
   if face = 5 then count5 = count5 + 1
   if face = 6 then count6 = count6 + 1
```

  - Each condition is exclusive from each other.

Conditionals

- List of conditions
  - Count the number of grades in a class.

```
reset countA to countF value to zero
for each student s in class list L do
    g = score of student s
    if g >= 85 then countA = countA + 1
    if g >= 70 then countB = countB + 1
    if g >= 55 then countC = countC + 1
    if g >= 40 then countD = countD + 1
    if g < 40 then countF = countF + 1
```

  - Each condition is not exclusive from each other!

Is this correct?

Conditionals

- List of conditions
  - Count the number of grades in a class.

```
reset countA to countF value to zero
for each student s in class list L do
    g = score of student s
    if g >= 85 then countA = countA + 1
    if 70 <= g < 85 then countB = countB + 1
    if 55 <= g < 70 then countC = countC + 1
    if 40 <= g < 55 then countD = countD + 1
    if g < 40 then countF = countF + 1
```

  - Each condition is now exclusive from each other!

Is this correct?

Conditionals

- Chained conditions
  - Count the number of grades in a class.

```
reset countA to countF value to zero
for each student s in class list L do
    g = score of student s
    if g >= 85 then countA = countA + 1
    else if g >= 70 then countB = countB + 1
    else if g >= 55 then countC = countC + 1
    else if g >= 40 then countD = countD + 1
    else if g < 40 then countF = countF + 1
```

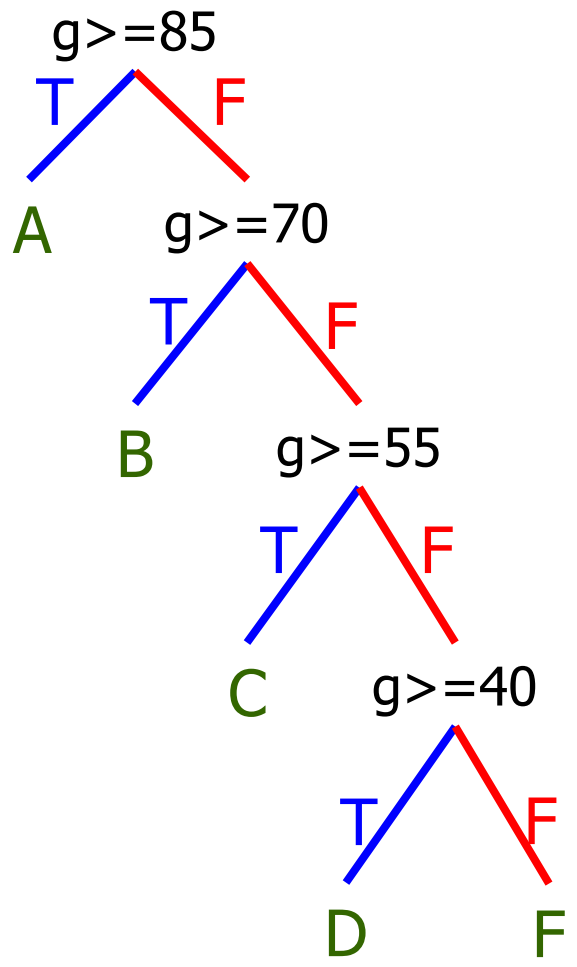
Is this correct?

  - Each condition is not exclusive from each other, but each in chain of conditions is exclusive!

Conditionals

- Chained conditions
  - In Python, else if can be replaced by elif

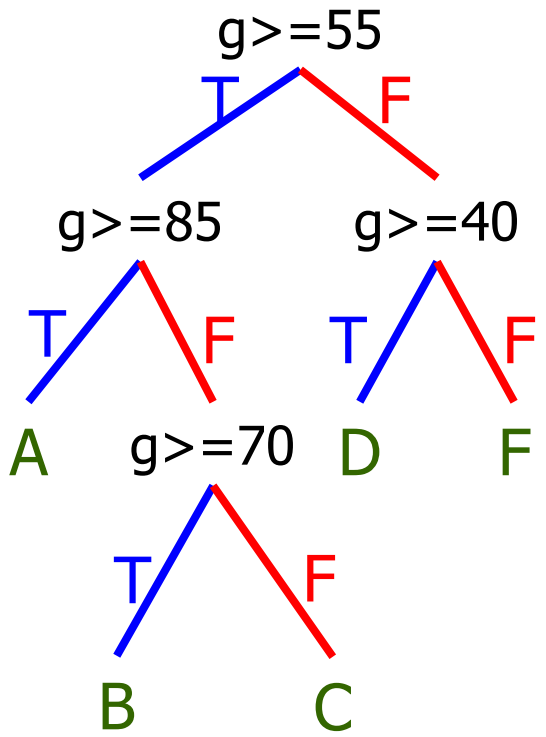
```
reset countA to countF value to zero
for each student s in class list L do
    g = score of student s
    if g >= 85 then countA = countA + 1
    elif g >= 70 then countB = countB + 1
    elif g >= 55 then countC = countC + 1
    elif g >= 40 then countD = countD + 1
    else countF = countF + 1
```



Nested/multi-way conditions

- Nested/multi-way conditions

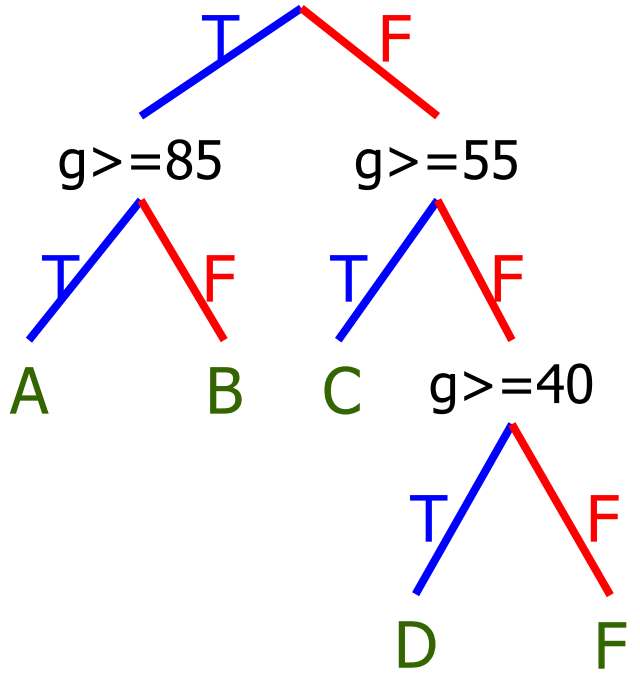
```
reset countA to countF value to zero
for each student s in class list L do
   g = score of student s
   if g >= 55 then # pass
            if g >= 85 then
                countA = countA + 1
            else
                if g >= 70 then countB = countB + 1
                else countC = countC + 1
   else
            if g >= 40 then countD = countD + 1
            else countF = countF + 1
```



Question 8

- Conditions can be reordered

```
reset countA to countF value to zero
for each student s in class list L do
   g = score of student s
   if g >= 70 then
          if g >= 85 then countA = countA + 1
          else countB = countB + 1
   else
          if g >= 55 then
              countC = countC + 1
          else
              if g >= 40 then countD = countD + 1
              else countF = countF + 1
```

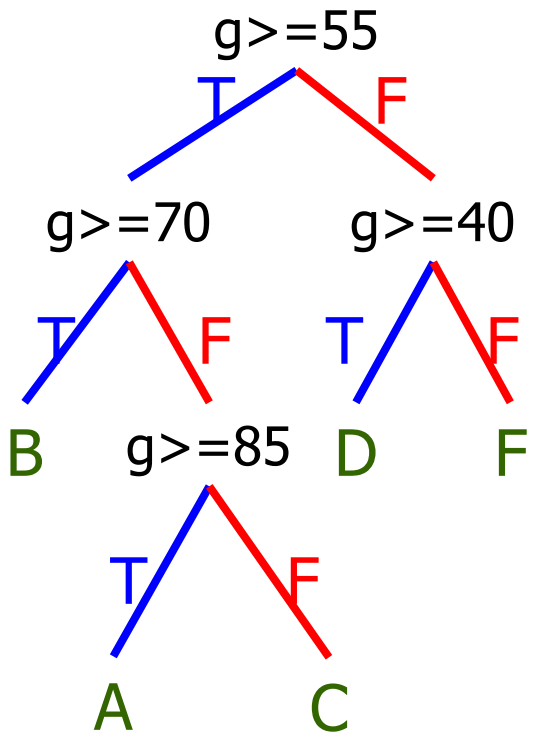


Is this pseudocode correct?

Question 9

- Conditions can be reordered

```
reset countA to countF value to zero
for each student s in class list L do
   g = score of student s
   if g >= 55 then # pass
          if g >= 70 then
              countB = countB + 1
          else
              if g >= 85 then countA = countA + 1
              else countC = countC + 1
   else
          if g >= 40 then countD = countD + 1
          else countF = countF + 1
```



Is this pseudocode correct?

Efficiency

- Which ordering of conditionals is the best?
- How do we determine which ordering is the best?
- We want to minimize the number of if-statements that are evaluated!
- But what metric?
  - Best-case
  - Worst-case
  - Average-case

Efficiency – Best-case

- Best-case number of if-statements
  - The student's grade **minimizes** the number of if-statements we need to evaluate.

if g >= 85 then countA = countA + 1 if 70 <= g < 85 then countB = countB + 1 if 55 <= g < 70 then countC = countC + 1 if 40 <= g < 55 then countD = countD + 1 if g < 40 then countF = countF + 1	if g >= 85 then countA = countA + 1 else if g >= 70 then countB = countB + 1 else if g >= 55 then countC = countC + 1 else if g >= 40 then countD = countD + 1 else countF = countF + 1

if g >= 55 then
  if g >= 85 then
  countA = countA + 1
  else
  if g >= 70 then countB = countB + 1
  else countC = countC + 1
else
  if g >= 40 then countD = countD + 1
  else countF = countF + 1
	if g >= 70 then
	  if g >= 85 then countA = countA + 1
	  else countB = countB + 1
	else
	  if g >= 55 then
	  countC = countC + 1
	  else
	  if g >= 40 then countD = countD + 1
	  else countF = countF + 1

Efficiency – Worst-case

- Worst-case number of if-statements
  - The student's grade **maximizes** the number of if-statements we need to evaluate.

if g >= 85 then countA = countA + 1 if 70 <= g < 85 then countB = countB + 1 if 55 <= g < 70 then countC = countC + 1 if 40 <= g < 55 then countD = countD + 1 if g < 40 then countF = countF + 1	if g >= 85 then countA = countA + 1 else if g >= 70 then countB = countB + 1 else if g >= 55 then countC = countC + 1 else if g >= 40 then countD = countD + 1 else countF = countF + 1

if g >= 55 then
  if g >= 85 then
  countA = countA + 1
  else
  if g >= 70 then countB = countB + 1
  else countC = countC + 1
else
  if g >= 40 then countD = countD + 1
  else countF = countF + 1
	if g >= 70 then
	  if g >= 85 then countA = countA + 1
	  else countB = countB + 1
	else
	  if g >= 55 then
	  countC = countC + 1
	  else
	  if g >= 40 then countD = countD + 1
	  else countF = countF + 1

Efficiency – Average-case

- Average-case number of if-statements
  - We can say that there is an equal number of all grades
  - **A=20%, B=20%, C=20%, D=20%, F=20%**

```
if g >= 85 then countA = countA + 1
if 70 <= g < 85 then countB = countB + 1
if 55 <= g < 70 then countC = countC + 1
if 40 <= g < 55 then countD = countD + 1
if g < 40 then countF = countF + 1
```

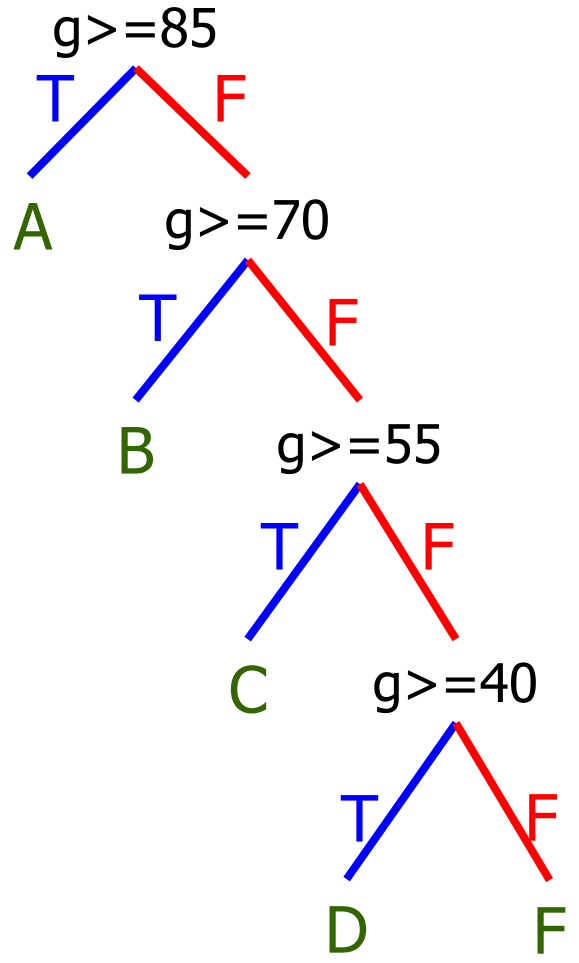
List of Conditions

Efficiency – Average-case

- Average-case number of if-statements
  - We can say that there is an equal number of all grades
  - **A=20%, B=20%, C=20%, D=20%, F=20%**

```
if g >= 85 then countA = countA + 1
else if g >= 70 then countB = countB + 1
else if g >= 55 then countC = countC + 1
else if g >= 40 then countD = countD + 1
else countF = countF + 1
```

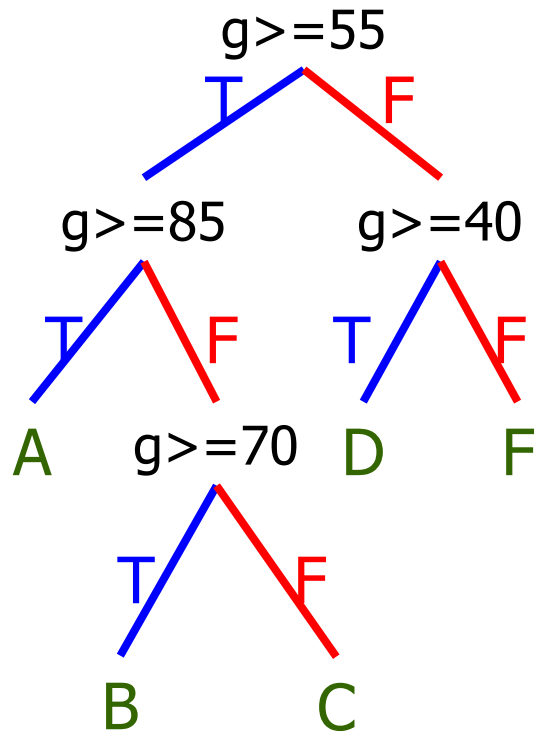
Chained Conditions



Efficiency – Average-case Question 10

- **What is the average number of if-statements evaluated per-student?**
  - We can say that there is an equal number of all grades
  - **A=20%, B=20%, C=20%, D=20%, F=20%**

```
if g >= 55 then
   if g >= 85 then
      countA = countA + 1
   else
      if g >= 70 then countB = countB + 1
      else countC = countC + 1
else
      if g >= 40 then countD = countD + 1
      else countF = countF + 1
```

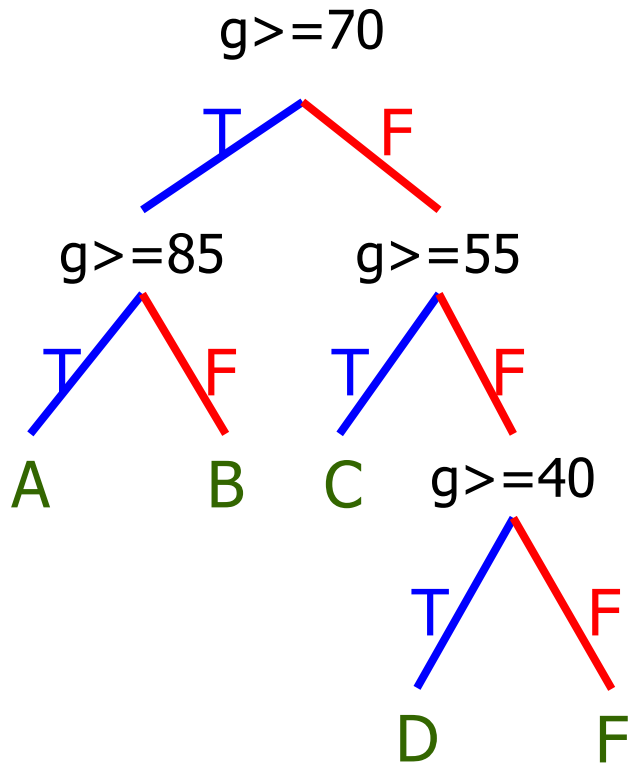


Nested Conditions 1

Efficiency – Average-case Question 11

- **What is the average number of if-statements evaluated per-student?**
  - We can say that there is an equal number of all grades
  - **A=20%, B=20%, C=20%, D=20%, F=20%**

```
if g >= 70 then
   if g >= 85 then countA = countA + 1
   else countB = countB + 1
else
   if g >= 55 then
      countC = countC + 1
   else
      if g >= 40 then countD = countD + 1
         else countF = countF + 1
```



Nested Conditions 2

Efficiency – Average-case

- Average-case number of if-statements
  - We can take the distribution of students from real-world data
  - **A=20%, B=50%, C=10%, D=10%, F=10%**

```
if g >= 85 then countA = countA + 1
if 70 <= g < 85 then countB = countB + 1
if 55 <= g < 70 then countC = countC + 1
if 40 <= g < 55 then countD = countD + 1
if g < 40 then countF = countF + 1
```

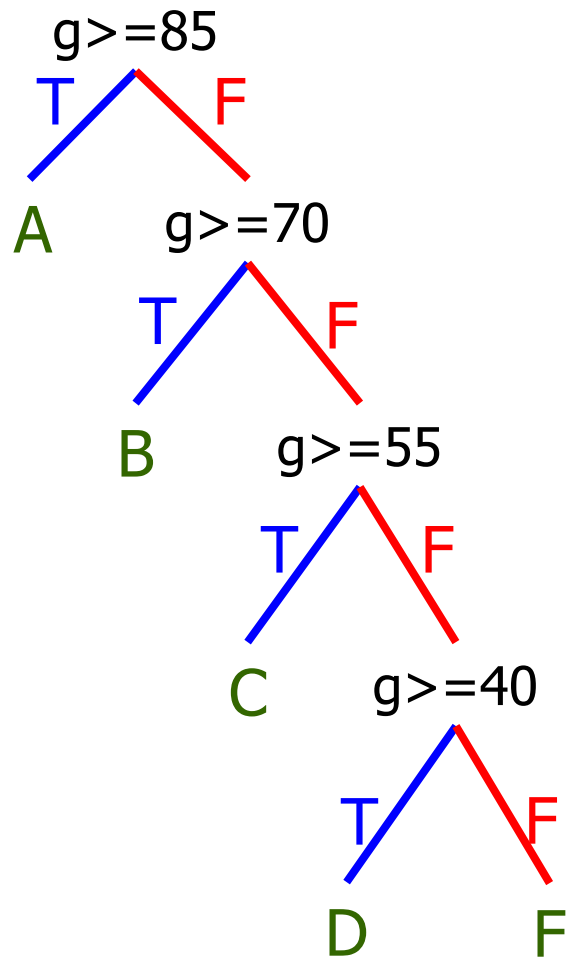
List of Conditions

Efficiency – Average-case

- Average-case number of if-statements
  - We can take the distribution of students from real-world data
  - **A=20%, B=50%, C=10%, D=10%, F=10%**

```
if g >= 85 then countA = countA + 1
else if g >= 70 then countB = countB + 1
else if g >= 55 then countC = countC + 1
else if g >= 40 then countD = countD + 1
else countF = countF + 1
```

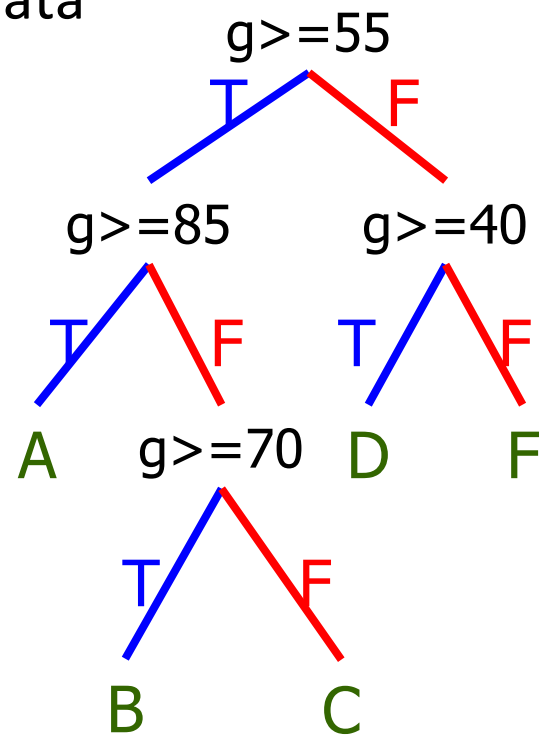
Chained Conditions



Efficiency – Average-case Question 12

- **What is the average number of if-statements evaluated per-student?**
  - We can take the distribution of students from real-world data
  - **A=20%, B=50%, C=10%, D=10%, F=10%**

```
if g >= 55 then
   if g >= 85 then
      countA = countA + 1
   else
      if g >= 70 then countB = countB + 1
      else countC = countC + 1
else
      if g >= 40 then countD = countD + 1
      else countF = countF + 1
```

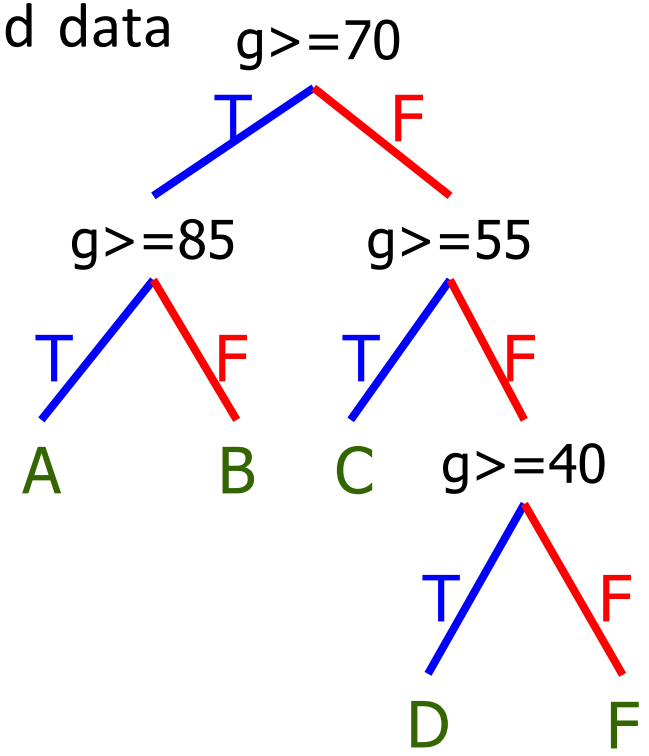


Nested Conditions 1

Efficiency – Average-case Question 13

- **What is the average number of if-statements evaluated per-student?**
  - We can take the distribution of students from real-world data
  - **A=20%, B=50%, C=10%, D=10%, F=10%**

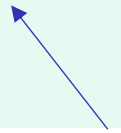
```
if g >= 70 then
   if g >= 85 then countA = countA + 1
   else countB = countB + 1
else
   if g >= 55 then
      countC = countC + 1
   else
      if g >= 40 then countD = countD + 1
         else countF = countF + 1
```



Nested Conditions 2

Efficiency - Summary

	List of conditions	Chained conditions	Nested conditions 1	Nested conditions 2
Best case	5	**1**	2	2
Worst case	5	4	**3**	**3**
Average case – equal count	5	2.8	**2.4**	**2.4**
Average case – realistic count	5	2.3	2.6	**2.2**



Can we beat this?

Efficiency

- **A=20%, B=50%, C=10%, D=10%, F=10%**

```
if 70 <= g < 85 then countB = countB + 1
else if g >= 85 then countA = countA + 1
else if g >= 55 then countC = countC + 1
else if g >= 40 then countD = countD + 1
else countF = countF + 1
```

- 0.5*1+0.2*2+0.1*3+0.1*4+0.1*5 = 2.1 < 2.2

Efficiency – Side-note

- In practice, “if 70 <= g < 85” takes two operations to evaluate.
  - It is evaluated as “if g>=70 and g < 85”
- “g >= 70” takes one operation to evaluate

- But for this lecture, we are counting if-statements, not evaluation operations!
  - “if 70 <= g < 85” counted as **one** if-statement

Break Time

- Put your laptops away
- Form groups of 4 to 8 students
- On a piece of paper, write at the top:
  - Group Name
  - Student Names
  - Student Numbers
- Make sure there is room for other writing on your paper

Alternate Grading System

- Most universities in Australia use the following grading system
  - High Distinction (HD): 85-100
  - Distinction (D): 75-84
  - Credit (C): 65-74
  - Pass (P): 50-64
  - Fail (F): 0-49

Group Challenge 1: 20 minutes

Alex roughly remembers that in his first year Electrical Engineering course, the grade distribution was:

HD (85-100)	D (75-84)	C (65-74)	P (50-64)	F (0-49)
10%	15%	25%	20%	30%

```
if g >= 85 then countA = countA + 1
else if g >= 70 then countB = countB + 1
else if g >= 55 then countC = countC + 1
else if g >= 40 then countD = countD + 1
else countF = countF + 1
```

^For your reference

- Write pseudocode to count the number of students in each grade band.
  - The variable g represents the student's grade
  - Use variables countHD, countD, countC, countP, countF
  - You can only use those variables and {if, else-if, then, >=, >, <=, <, ==, !=, =, +, 1}
  - "if a <= g < b" will be counted as **one** if-statement

No Laptops Allowed!

- Draw a decision tree which represents the pseudocode.
- Showing your working out, do average-case analysis to find the average number of if-statements evaluated per-student for this grade distribution.
- Submit your sheet to the front when you are done.
- **Team with lowest avg wins. Pseudocode+Tree+Analysis must be correct. Tie-break in favour of earlier submission. Top two teams win a prize.**

Break Time

- On **two** pieces of paper, write at the top:
  - Group Name
  - Student Names
  - Student Numbers
- Make sure there is room for other writing on your paper

Group Challenge 2: 15 minutes

Now, the percentages are randomly generated **after** your pseudocode is submitted.

HD (85-100)	D (75-84)	C (65-74)	P (50-64)	F (0-49)
RNG%	RNG%	RNG%	RNG%	RNG%

- On **both pieces of paper,** write pseudocode and a corresponding decision tree to count the number of students in each grade band.
  - **The same pseudocode and decision tree must be on both pieces of paper.**
  - The variable g represents the student's grade
  - Use variables countHD, countD, countC, countP, countF
  - You can only use those variables and {if, else-if, then, >=, >, <=, <, ==, !=, =, +, 1}
  - "if a <= g < b" will be counted as **one** if-statement

No Laptops Allowed!

```
if g >= 85 then countA = countA + 1
else if g >= 70 then countB = countB + 1
else if g >= 55 then countC = countC + 1
else if g >= 40 then countD = countD + 1
else countF = countF + 1
```

^For your reference

- Submit **one of your sheets** to the front when you are done.
- When the time is up, I will randomly generate the grade distribution. You will then do the average-case analysis on your **other sheet of paper**. **Do not submit this paper (yet).**
- **Team with lowest avg wins. Pseudocode+Tree+Analysis must be correct. Tie-break in favour of earlier Sheet1 submission. Top two teams win a prize.**

Group Challenge is now over

You can take your laptop out

Counting if-statement evaluations in Python

```
def countif_2():
    grade_list = [90,70,50,30,60]
    countA, countB, countC, countD, countF = 0, 0, 0, 0, 0
    if_count = 0
    for g in grade_list:
        if g >= 85:
            if_count = if_count+1
            countA = countA + 1
        elif g >= 70:
            if_count = if_count+2
            countB = countB + 1
        elif g >= 55:
            if_count = if_count+3
            countC = countC + 1
        elif g >= 40:
            if_count = if_count+4
            countD = countD + 1
        else:
            if_count = if_count+4
            countF = countF + 1
    print("If-2: Grades",grade_list)
    print("Total if",if_count,"Average",if_count/len(grade_list))
```

Homework: Try implement this for the other lecture examples!

Revisiting List of Conditions

- Consider this again:

```
reset countA to countF value to zero
for each student s in class list L do
    g = score of student s
    if g >= 85 then countA = countA + 1
    if 70 <= g < 85 then countB = countB + 1
    if 55 <= g < 70 then countC = countC + 1
    if 40 <= g < 55 then countD = countD + 1
    if g < 40 then countF = countF + 1
```

- Conditions are mutually exclusive.
  - Once a count is added, there is no need to check for other if-cases. We could stop the checking and go to next student.
  - This can be done with **`continue`** in Python/C.

Counting Inside a Program

```
for g in grade_list:
    if g >= 85:
        if_count = if_count+1
        countA = countA + 1
        continue
    if_count = if_count+1
    if 70 <= g < 85:
        if_count = if_count+1
        countB = countB + 1
        continue
    if_count = if_count+1
    if 55 <= g < 70:
        if_count = if_count+1
        countC = countC + 1
        continue
    if_count = if_count+1
    if 40 <= g < 55:
        if_count = if_count+1
        countD = countD + 1
        continue
    if_count = if_count+1
    if g < 40:
        if_count = if_count+1
        countF = countF + 1
        continue
    if_count = if_count+1
```

`continue` will go directly to the next round of loop

Homework: Try implement this too!

New Consultation Hour!

- Friday 3:30 – 4:30 pm at LibCafe


That's all Folks!