Nested Branching & Logical Operators

Lecture 4 (Feb 2)
CS100M – Spring 2006

Announcements

- Project 2
  - Due Thursday, Feb 16
  - Should appear online by this weekend
- For this next week, section will be in the classroom instead of the lab

Topics

- Recall previous lecture
  - Used if-else-end construct to find min of two values: q(L) & q(R) where q is a quadratic polynomial
- Plans for today
  - More complicated branching
  - Logical operators

Goal

- Create a Matlab program to determine the minimum value of

$$q(x) = x^2 + bx + c$$

  in the interval [L, R]
- We know how to do this using Calculus
  - The answer has to be one of $q(x_c)$, q(L), or q(R) where x_c is the critical point (where the derivative is zero)
  - But we use $q(x_c)$ only if x_c is in [L, R]

Algorithm Outline

- Compute x_c
- If $x_c \in [L, R]$
  - Answer is $q(x_c)$
- Otherwise
  - Answer is min of q(L) and q(R)

We already know how to do this

Algorithm (with More Detail)

- Compute x_c
- If $L \leq x_c \leq R$
  - Answer is $q(x_c)$
- Otherwise
  - Compute qL = q(L); compute qR = q(R)
  - If qL < qR
    - Answer is qL
  - Otherwise
    - Answer is qR

We have an if-construct inside another if-construct

Program Fragment

```
% Determine min value of q(x) = x^2 + b*x + c
% in the interval [L, R]
xc = - b/2;          % Compute xc
if (L <= xc && xc <= R)
    minValue = xc^2 + b*xc + c;
else                 % Compute min of q(L) and q(R)
    qL = L^2 + b*L + c;
    qR = R^2 + b*R + c;
    if (qL < qR)
        minValue = qL;
    else
        minValue = qR;
    end
end
fprintf('Min value is %f\n', minValue)
```

Things to Note

- An if-construct can appear within a branch just like any other kind of statement
- Matlab (and most other programming languages) treat comparison operators as *binary* operators
  - Thus some kinds of standard math notation do not work in a Matlab program
    - Math: If $1 < x < 10$ then…
    - Matlab: if `(1 < x && x < 10)` …
- Indentation helps make the program readable, but Matlab doesn't enforce indentation rules
  - Your projects are graded on both correctness and style
    - Appropriate indentation is necessary to achieve a good style grade
  - The Matlab Editor helps with the indentation
    - You can override this, but you shouldn't

Logical And

- How do we check if x_c is in [L, R]?
  - We check $L \leq x_c$ and $x_c \leq R$
  - In our code: `(L <= xc && xc <= R)`
- Rules for logical and:

x	y	x and y
F	F	F
F	T	F
T	F	F
T	T	T

Logical Or

- Alternately, we could check if x_c is outside of [L, R]
  - We check $x_c \leq L$ or $R \leq x_c$
  - In our code: `(xc <= L || R <= xc)`
- Rules for logical or:

x	y	x or y
F	F	F
F	T	T
T	F	T
T	T	T

Logical Operators

- Logical and: &&
- Logical or: ||
- Logical not: ~
- Matlab uses 0 for false and nonzero for true
  - Uses 1 for true when Matlab generates it, but will take any nonzero as true in a logical expression
  - Matlab also has predefined logical constants:
    - false (= 0) and true (= 1)

Comparison Operators

- Equal ==
- Not equal ~=
- Less than <
- Greater than >
- Less than or equal <=
- Greater than or equal >=
- Each of these operators produces a boolean result (i.e., the result is either true or false)
- Note use of == to compare for equality

Some Built-In Functions

- Most standard mathematical functions are available
  - When in doubt type `help functionName` in the Command Window
- Trigonometric functions (using radians, not degrees)
  - sin
  - cos
  - tan
  - asin (inverse sin)
  - acos (inverse cos)
  - atan (inverse tan)
- Log, exponential functions
  - exp (exponential)
  - log (natural logarithm)
  - log10 (base-10 logarithm)
  - log2 (base-2 logarithm)
  - Also, `x^p` computes x^p
- Functions for integer computation
  - floor
  - ceil
  - round
  - fix
  - mod
- A few more: max, min, abs

floor

p = floor(x)

- p is assigned the largest integer less than or equal to x

floor(-3.5) has the value -4
floor(3.5) has the value 3
floor(5) has the value 5
floor(3.2) has the value 3
floor(3.7) has the value 3

ceil

p = ceil(x)

- p is assigned the smallest integer greater than or equal to x

ceil(-3.5) has the value -3
ceil(3.5) has the value 4
ceil(5) has the value 5
ceil(3.2) has the value 4
ceil(3.7) has the value 4

round

p = round(x)

- p is assigned the integer that is closest to x
  - In case of a tie, use the integer that is farther from 0

round(-3.5) has the value -4
round(3.5) has the value 4
round(5) has the value 5
round(3.2) has the value 3
round(3.7) has the value 4

fix

p = fix(x)

- p is assigned the closest integer between 0 and x (i.e., round toward 0)

fix(-3.5) has the value -3
fix(3.5) has the value 3
fix(5) has the value 5
fix(3.2) has the value 3
fix(3.7) has the value 3

mod

r = mod(p, q)

- r is assigned the remainder when we divide p by q

mod(5, 2) has the value 1
mod(704, 10) has the value 4
mod(30, 7) has the value 2

Boolean Expression Example

- To test if x is divisible by both 3 and 5

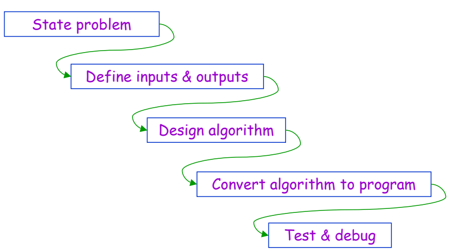
```
if (mod(x, 3) == 0 && mod(x, 5) == 0)
    disp('Divisible by both')
else
    disp('Not divisible by both')
end
```

Another Boolean Expression Example

- To test if integer y represents a Leap Year
  - Year y is a Leap Year if
    - It is divisible by 4
    - Exception: century years are *not* Leap Years
    - Exception: years divisible by 400 *are* Leap Years
- Resulting code fragment

```
if mod(y,400) == 0 || ( mod(y,4) == 0 && mod(y,100) ~= 0 )
    fprintf('%.0f is a Leap Year\n', y)
else
    fprintf('%.0f is not a Leap Year\n', y)
end
```

Creating a Program



- An algorithm is an *idea*
- To use an algorithm you must choose a programming language and *implement* the algorithm

Revisiting the Min-Finding Program

```
% Determine min value of q(x) = x^2 + b*x + c
% in the interval [L, R]
xc = - b/2;         % Compute xc
if (L <= xc && xc <= R)
    minValue = xc^2 + b*xc + c;
else                % Compute min of q(L) and q(R)
    minValue = min(L^2 + b*L + c, R^2 + b*R + c);
end
fprintf('Min value is %f\n', minValue)
```

Playing with Comparisons

- Suppose x has the value 5
  - What is the result of typing
    `x < 10`
    in the Matlab Command Window?
  - What is the result of typing
    `6 < x`
    in the Matlab Command Window?
  - What is the result of typing
    `6 < x < 10`
    in the Matlab Command Window?