

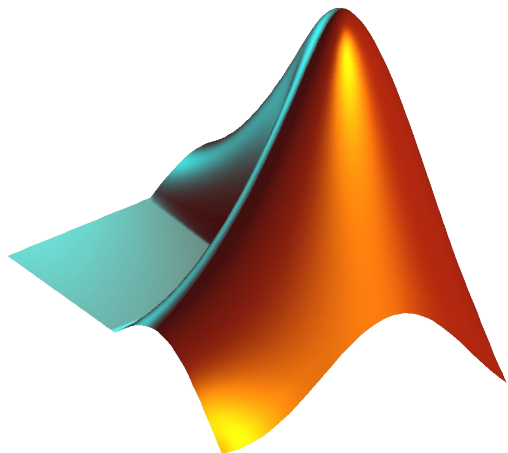
CS 1112 Introduction to Computing Using MATLAB

Instructor: Dominic Diaz

Website:

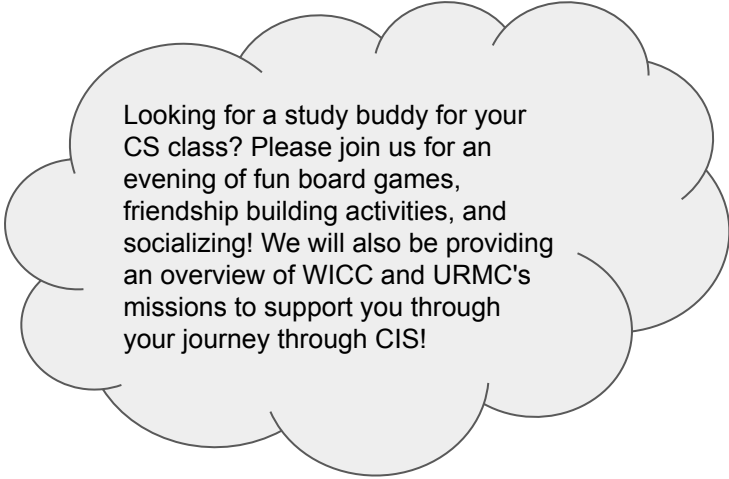
<https://www.cs.cornell.edu/courses/cs1112/2022fa/>

Today: Iteration, for loops



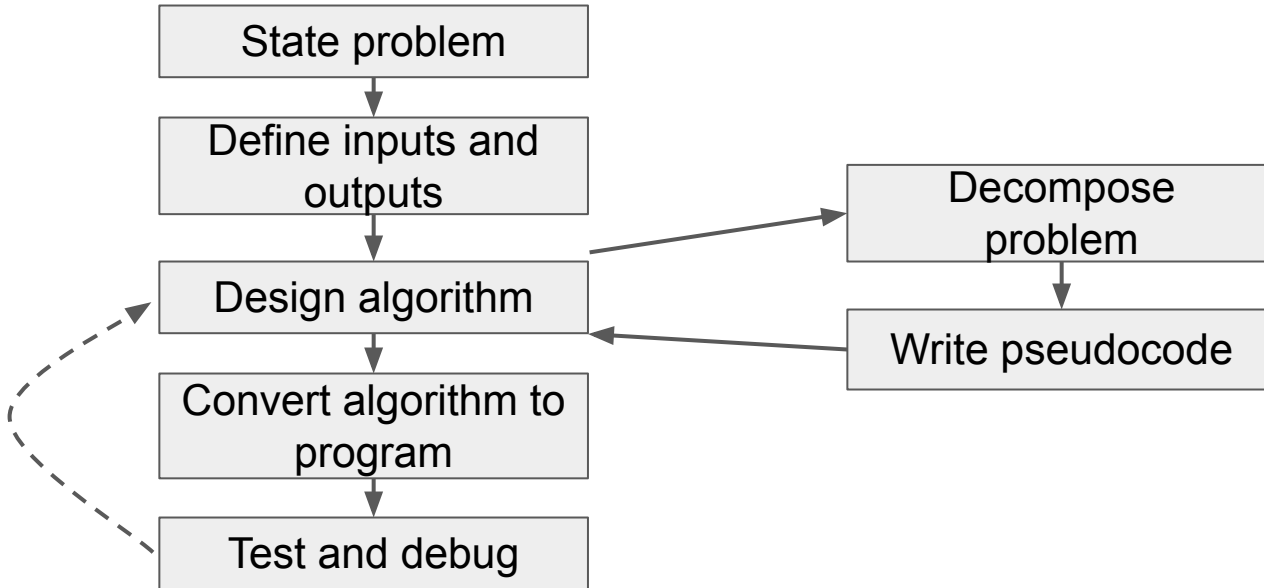
Agenda and announcements

- Last time
 - Conditionals and nested conditionals
- This time
 - Iteration using **for**
- Announcements
 - URMC x WICC Partner Finding Social
 - Sept 6th 4-6 PM in Upson lounge
 - Hint on Project 1 problem 2 posted
 - Project 1 due TOMORROW at 11 PM!
 - Late submission accepted until Thursday at 11 PM with 10% deduction
 - Poll Everywhere will be 75% completion, 25% correctness



Looking for a study buddy for your CS class? Please join us for an evening of fun board games, friendship building activities, and socializing! We will also be providing an overview of WICC and URMC's missions to support you through your journey through CIS!

Top-Down Design



An **algorithm** is a set of instructions to solve a particular problem. To **implement** an algorithm you must choose a programming language and code the algorithm in that language.

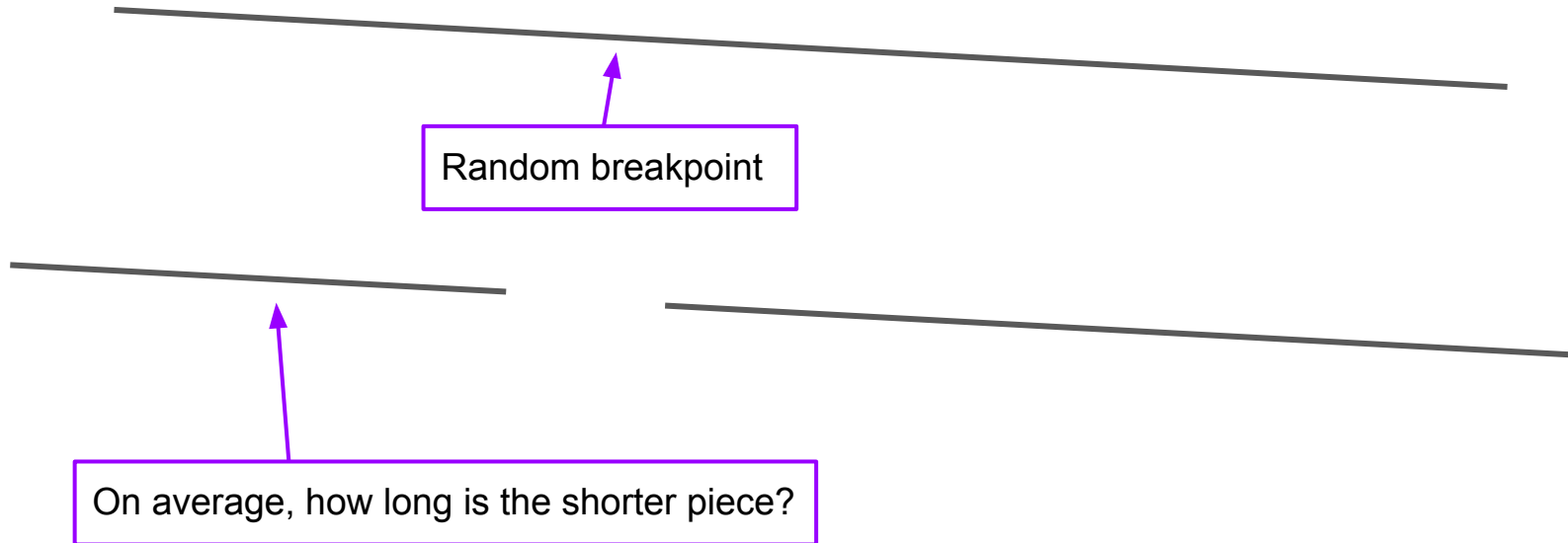
Example Question

A stick of unit length is split into two pieces. The breakpoint is randomly selected. On average, how long is the shorter piece?

How can we solve this question?

- Physical experiments
- Thought experiment (mathematical analysis)
- Computational experiment (simulation)

A stick of unit length is split into two pieces. The breakpoint is randomly selected. On average, how long is the shorter piece?



Solution using simulation

Simulation: use code to imitate a physical experiment

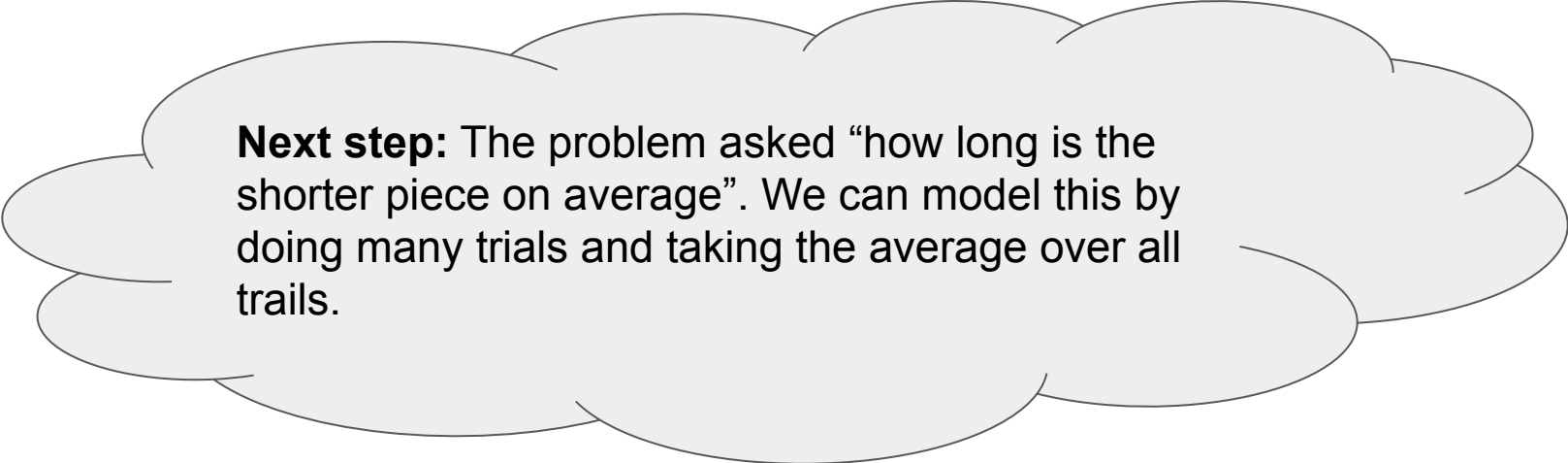
```
% one trial of the experiment
breakPt = rand();
if breakPt < 0.5
    shortPiece = breakPt;
else
    shortPiece = 1-breakPt;
end
```

Using shortcut: `min()`

```
% one trial of the experiment
```

```
breakPt = rand();
```

```
shortPiece = min(breakPt, 1-breakPt);
```



Next step: The problem asked “how long is the shorter piece on average”. We can model this by doing many trials and taking the average over all trials.

Algorithm: repeat experiment n times

Repeat n times

```
% one trial of the experiment
```

```
breakPt = rand();
```

```
shortPiece = min(breakPt, 1-breakPt);
```

Take average

Print result

Code to calculate average length of shorter stick

```
n = 10000;           % number of trials
total = 0;           % accumulated length of shorter sticks
                    % so far

for k = 1:1:n
    % one trial of the experiment
    breakPt = rand();
    shortPiece = min(breakPt, 1-breakPt);
    total = total + shortPiece;
end

aveLength = total/n;
fprintf('Average length is %f \n', aveLength);
```

Syntax of the `for` loop

```
for [var] = [startVal]:[stepSize]:[endVal]
```

```
    [statements to be executed multiple times]
```

```
end
```

Loop header specifies all the values that the index variable will take on, one for each pass of the loop. In the bottom example, `k = 1:1:n` means `k` will take on values 1, 2, 3, ..., n, one at a time.

```
for k = 1:1:n
    % one trial of the experiment
    breakPt = rand();
    shortPiece = min(breakPt, 1-breakPt);
    total = total + shortPiece;
end
```

For loop examples

```
for k = 2:2:6  
    disp(k);  
end
```

k takes on values 2, 4, 6
Start at 2, end at 6, step size of 2

```
for k = 5:10  
    disp(k);  
end
```

k takes on values 5, 6, 7, 8, 9, 10
If only 2 numbers provided, default step size = 1

```
for k = 0:-2:-7  
    disp(k);  
end
```

k takes on values 0, -2, -4, -6
End number specifies end bound

```
for k = 5:2:1  
    disp(k);  
end
```

k takes on values []
If end < start, loop body won't execute

Example: “Accumulate” a solution

```
% Average 10 numbers from user input
n = 10;           % number of input values
total = 0;       % current sum (initialized to zero)

for k = 1:n
    % input number and process input value
    num = input('Enter a number: ');
    total = total + num;
end

ave = total/n;   % average of inputted numbers
fprintf('Average of inputted numbers is %f\n', ave);
```

Accumulator variable

```
% Average 10 numbers from user input
n = 10; % number of data values
total = 0; % current sum (initialized to zero)

for k = 1:n
    % input number and process input value
    num = input('Enter a number: ');
    total = total + num;
end

ave = total/n; % average of inputted numbers
fprintf('Average of inputted numbers is %f\n', ave);
```