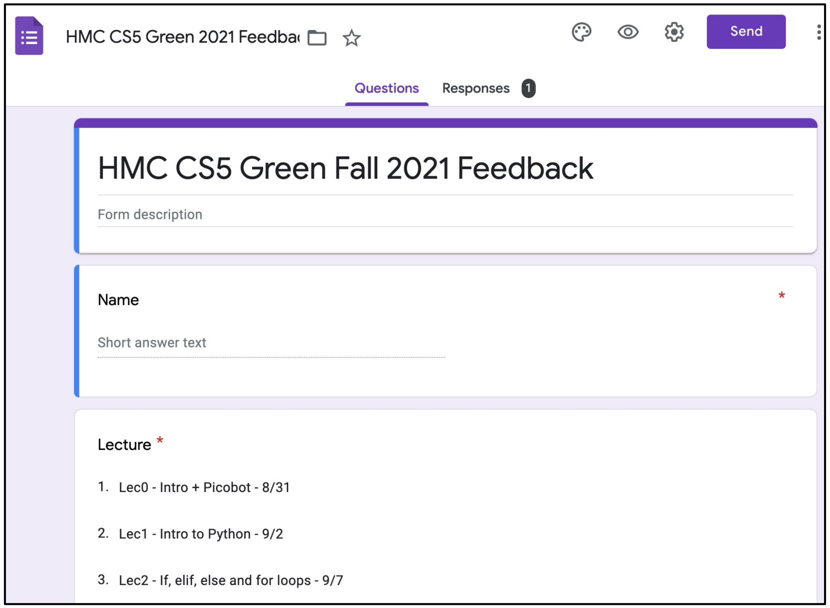
Welcome to CS5 Green

Two handouts today…

- Lecture notes
- Syllabus

Two online surveys…

- Intro survey
- Lecture feedback form



HMC CS5 Green Fall 2021 Feedback

Form description

Name *

Short answer text

Lecture *

1. Lec0 - Intro + Picobot - 8/31
2. Lec1 - Intro to Python - 9/2
3. Lec2 - If, elif, else and for loops - 9/7

Computing in the context of biological problems

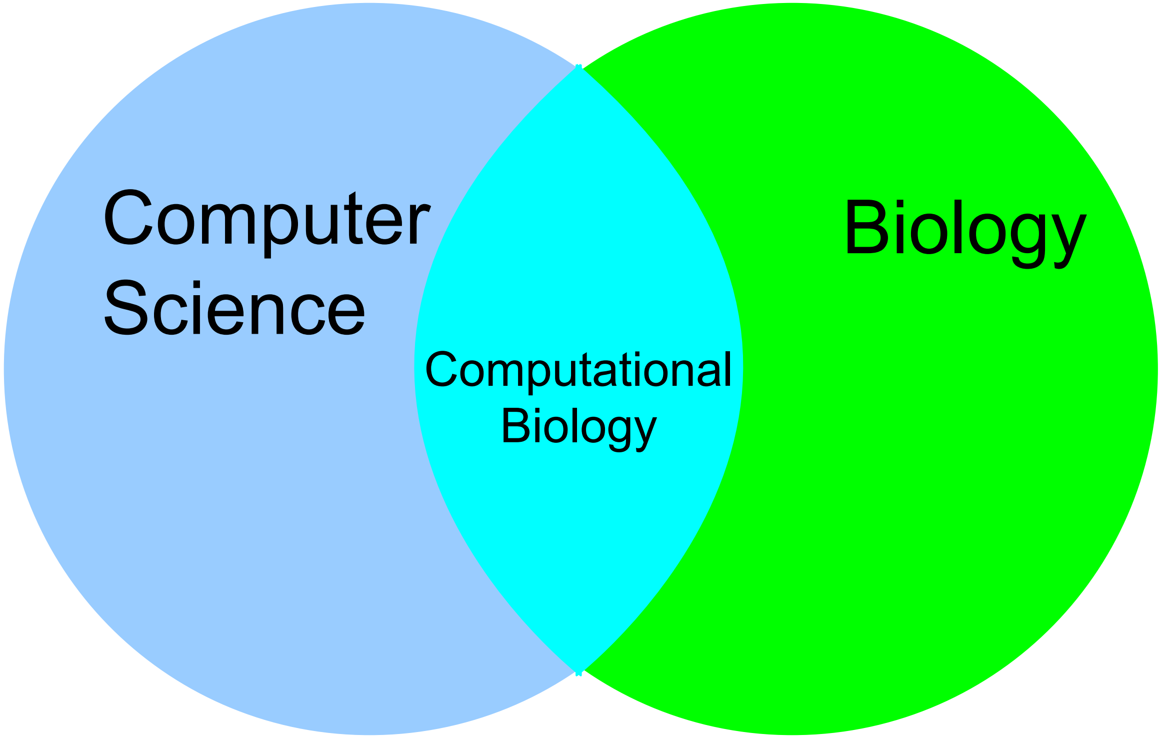
Are sex determination systems in birds and mammals related?

How does salmonella cause disease?

How are Neanderthals related to modern humans?

Neanderthal Museum, Mettman Germany

http://foodtestinguae.blogspot.com/
http://humanorigins.si.edu/evidence/human-fossils/fossils/la-chapelle-aux-saints

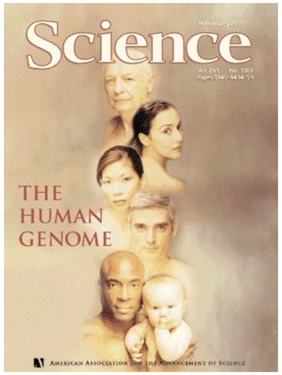


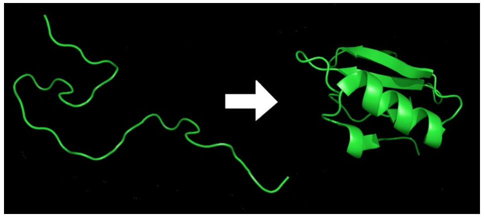
Computer Science

Biology

Computational Biology

Journal of Computational Biology
A Journal of Computational Molecular Cell Biology
Bioinformatics
PLoS COMPUTATIONAL BIOLOGY
a peer-reviewed open-access journal published by the the Public Library of Science
gcagtgtc0100001001010011





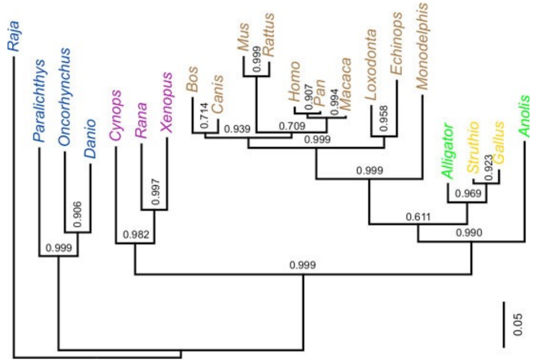
RESEARCH ARTICLE OPEN

Identifying the Important HIV-1 Recombination Breakpoints

Article | Metrics | Related Content | Comments: 0

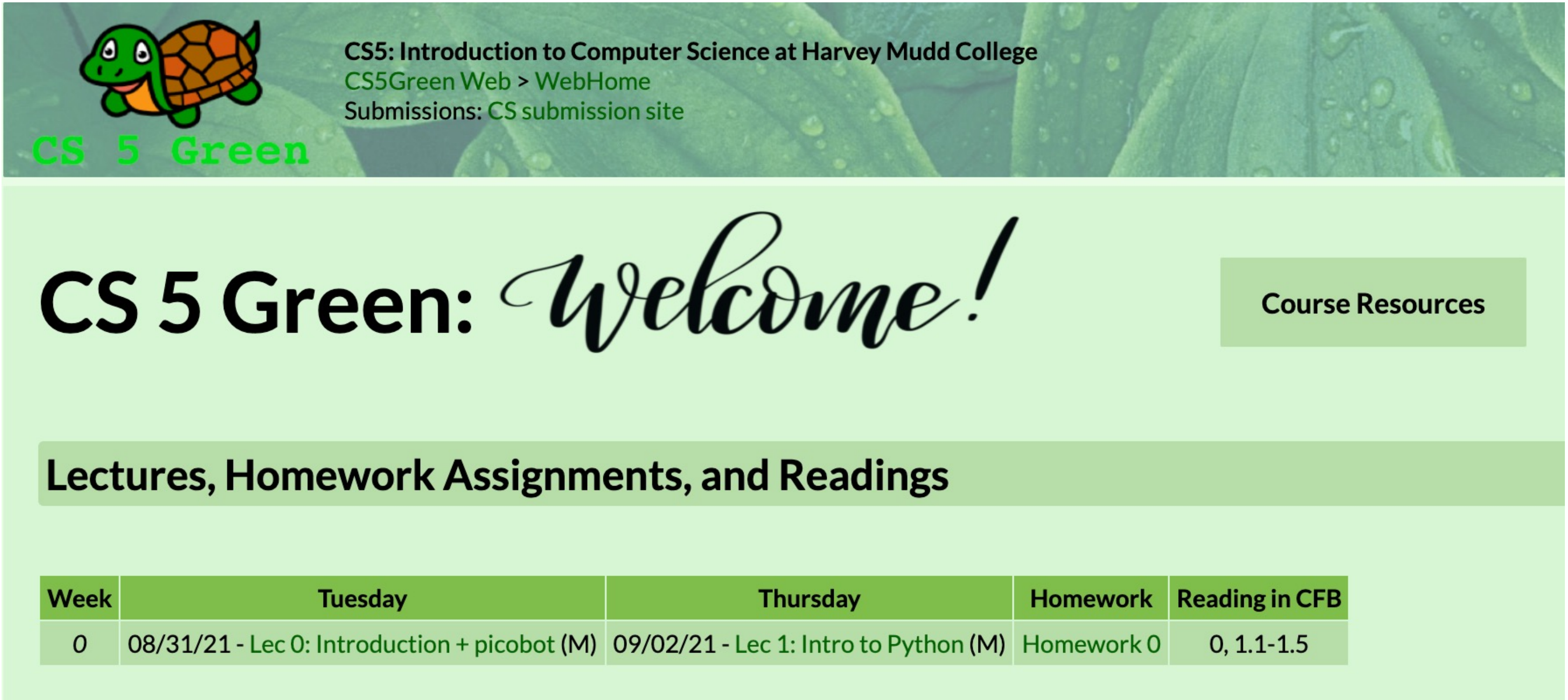
John Archer[1#], John W. Pinney[1#], Jun Fan[1], Etienne Simon-Loriere[2], Eric J. Arts[3], Matteo Negroni[2,4],

To add a note
some text. Hid



Course Website

www.cs.hmc.edu/cs5green



CS 5 Green

CS5: Introduction to Computer Science at Harvey Mudd College

CS5Green Web > WebHome

Submissions: CS submission site

CS 5 Green: *Welcome!*

Lectures, Homework Assignments, and Readings

Week	Tuesday	Thursday	Homework	Reading in CFB
0	08/31/21 - Lec 0: Introduction + picobot (M)	09/02/21 - Lec 1: Intro to Python (M)	Homework 0	0, 1.1-1.5

Syllabus in a Nutshell

Lectures: Tuesday and Thursday, 9:35-10:50, Shan 2460

Labs: Thursdays 1:15-3:15 PM in Shan B442
Recommended (and incentivized), but not required

Office hours and grutoring hours on the website!

Piazza Q&A system

piazza

Why Homework is Better with Friends

52 Admit it, you had the same dumb question.

39 When the prof says, "Any questions?" there are actually questions.

29 When you're totally lost, someone less lost can be a lifesaver.

WELCOME TO PIAZZA'S STUDENT QUAD FOR

Harvey Mudd College

Piazza is a free, online question and answer platform for classes. Ask questions (even anonymously) and get answers from classmates and instructors fast.

Homework

- Lab Problem
- Several additional homework problems

- Pair programming encouraged on some problems
- Due Mondays at 11:59 PM (Gradescope)

- Three CS 5 Greenbacks (aka "Euros")

Syllabus in a Nutshell

Pair Programming Policy: For some questions, you are (optionally) allowed to work as a pair. In a pair you should always program together and switch every 30 minutes.

Honor Code Policy: Other than pair programming, discussions OK, sharing or searching for code not permitted.

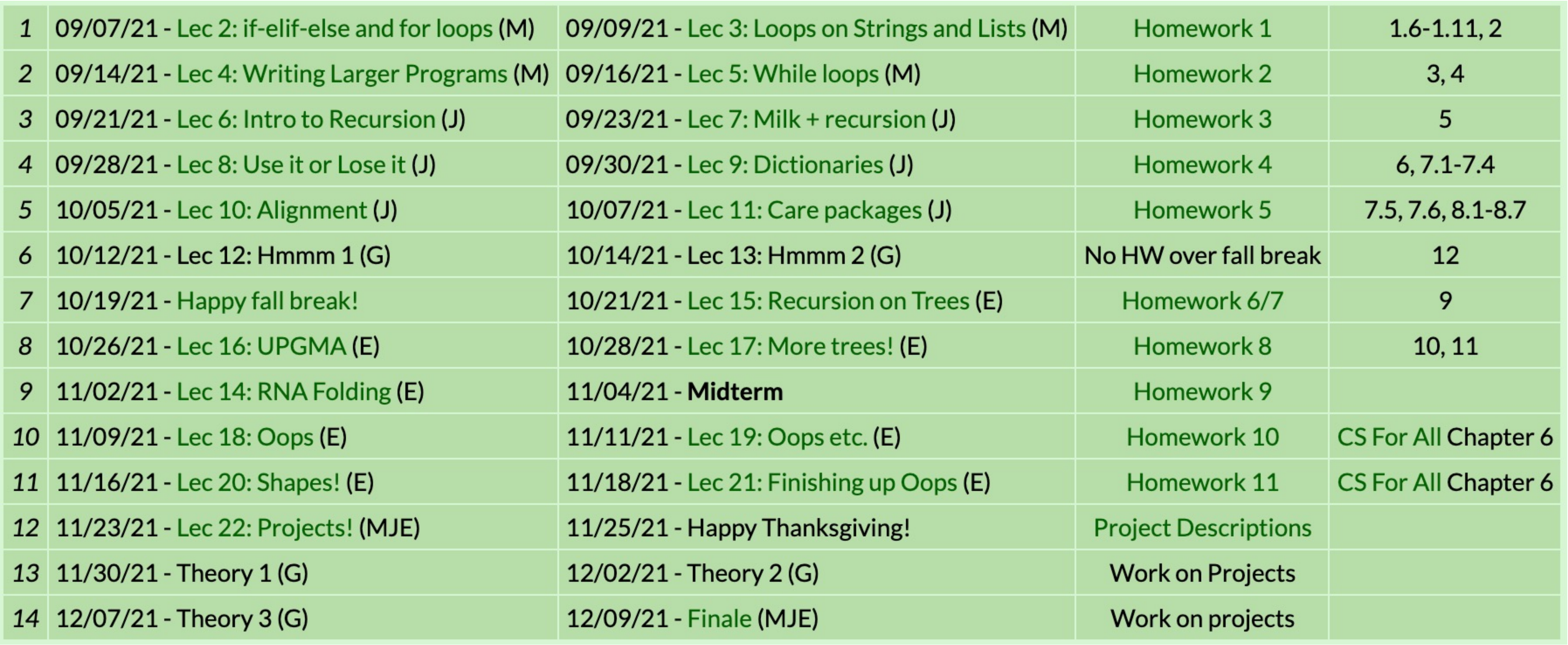
Grading:

- Homework + Final Project: 65%
- Midterm: (Thursday, Nov 4 in-class): 10%
- Final Exam: (Tuesday, December 14, 2pm-5pm): 20%
- Participation/worksheets: 5% (missing up to 3 is OK)

To pass CS 5, one must have a passing grade on all components (Homework, Exams, Participation)

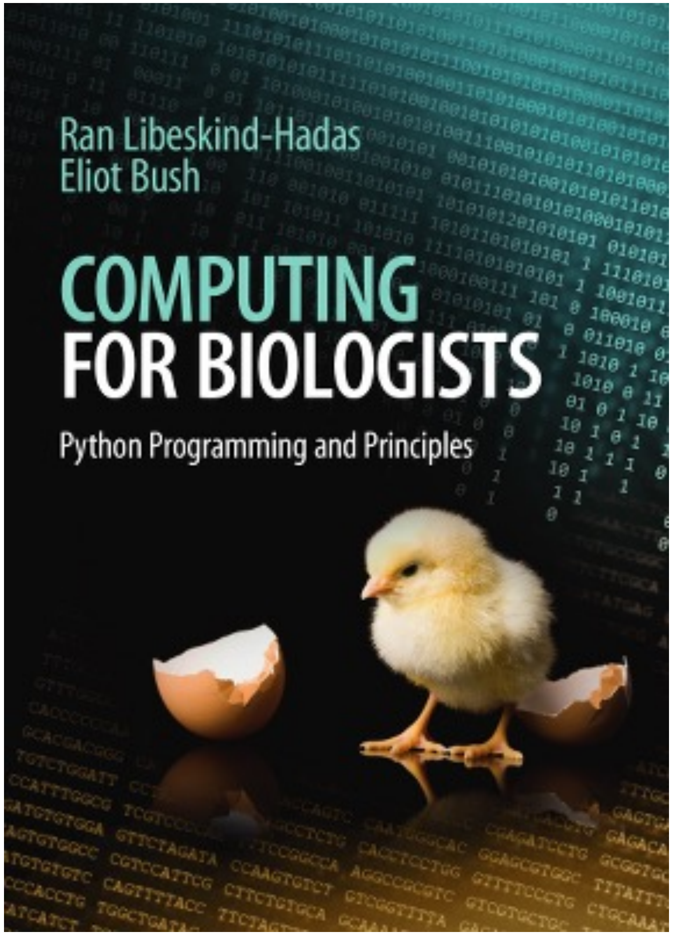
Pass fail vs. graded

Looking Ahead
www.cs.hmc.edu/cs5green



1	09/07/21 - Lec 2: if-elif-else and for loops (M)	09/09/21 - Lec 3: Loops on Strings and Lists (M)	Homework 1	1.6-1.11, 2
2	09/14/21 - Lec 4: Writing Larger Programs (M)	09/16/21 - Lec 5: While loops (M)	Homework 2	3, 4
3	09/21/21 - Lec 6: Intro to Recursion (J)	09/23/21 - Lec 7: Milk + recursion (J)	Homework 3	5
4	09/28/21 - Lec 8: Use it or Lose it (J)	09/30/21 - Lec 9: Dictionaries (J)	Homework 4	6, 7.1-7.4
5	10/05/21 - Lec 10: Alignment (J)	10/07/21 - Lec 11: Care packages (J)	Homework 5	7.5, 7.6, 8.1-8.7
6	10/12/21 - Lec 12: Hmmm 1 (G)	10/14/21 - Lec 13: Hmmm 2 (G)	No HW over fall break	12
7	10/19/21 - Happy fall break!	10/21/21 - Lec 15: Recursion on Trees (E)	Homework 6/7	9
8	10/26/21 - Lec 16: UPGMA (E)	10/28/21 - Lec 17: More trees! (E)	Homework 8	10, 11
9	11/02/21 - Lec 14: RNA Folding (E)	11/04/21 - **Midterm**	Homework 9	
10	11/09/21 - Lec 18: Oops (E)	11/11/21 - Lec 19: Oops etc. (E)	Homework 10	CS For All Chapter 6
11	11/16/21 - Lec 20: Shapes! (E)	11/18/21 - Lec 21: Finishing up Oops (E)	Homework 11	CS For All Chapter 6
12	11/23/21 - Lec 22: Projects! (MJE)	11/25/21 - Happy Thanksgiving!	Project Descriptions	
13	11/30/21 - Theory 1 (G)	12/02/21 - Theory 2 (G)	Work on Projects	
14	12/07/21 - Theory 3 (G)	12/09/21 - Finale (MJE)	Work on projects	

Textbook



Ran Libeskind-Hadas
Eliot Bush

COMPUTING FOR BIOLOGISTS

Python Programming and Principles

OAK (Occasionally Asked Kweschens)

Q: Will I learn as much CS here as I would in CS 5 Gold?

A: Yes!

Q: Are there other courses combining CS and Bio at Mudd

A: Yes! Bio 52, MCB118b, Bio 188, and a whole major (Mathematical and Computational Biology)

Our first programming language

Python

General-purpose language

you might see 50% by the end of the term

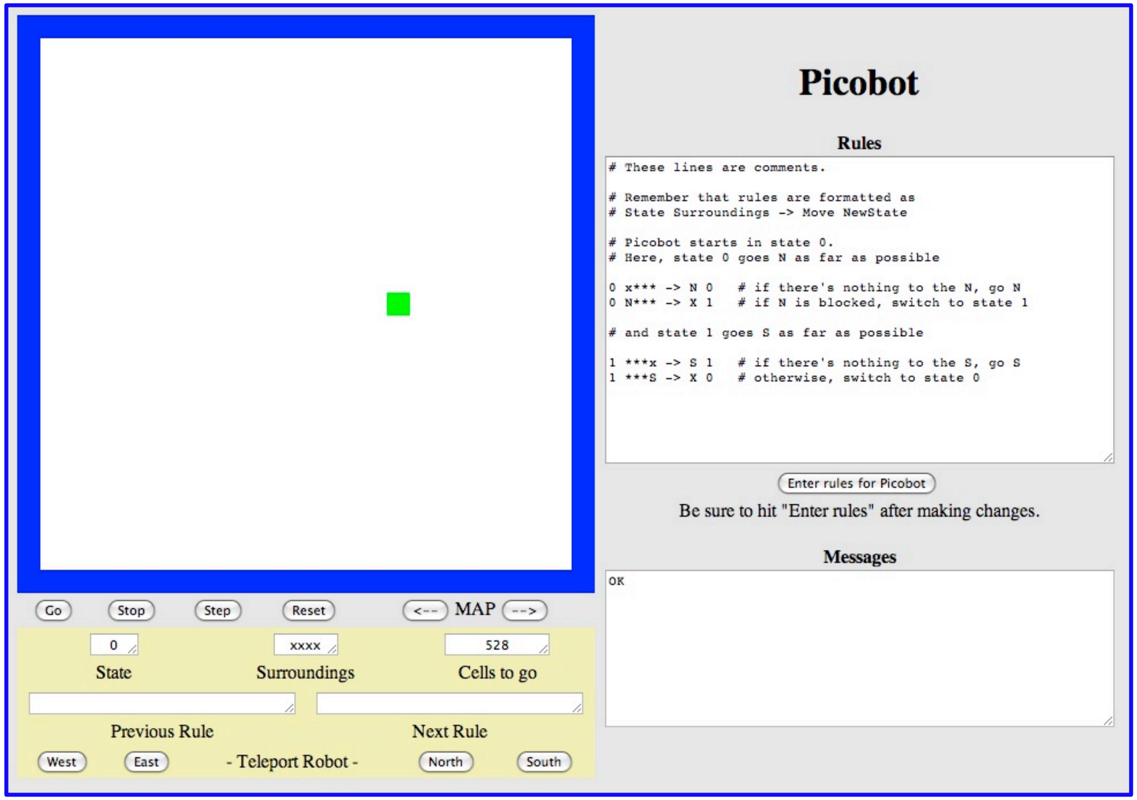
even then, <1% of its libraries!

Picobot

Special-purpose language

you'll see 100% in the next 10 minutes

Picobot!



```
# These lines are comments.

# Remember that rules are formatted as
# State Surroundings -> Move NewState

# Picobot starts in state 0.
# Here, state 0 goes N as far as possible

0 x*** -> N 0   # if there's nothing to the N, go N
0 N*** -> X 1   # if N is blocked, switch to state 1

# and state 1 goes S as far as possible

1 ***x -> S 1   # if there's nothing to the S, go S
1 ***S -> X 0   # otherwise, switch to state 0
```

Be sure to hit "Enter rules" after making changes.

Messages: OK

State: 0

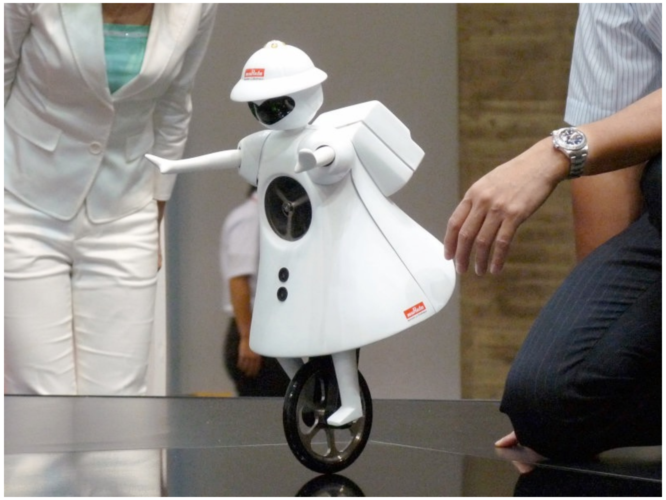
Surroundings: xxxx

Cells to go: 528

The Picobot simulator

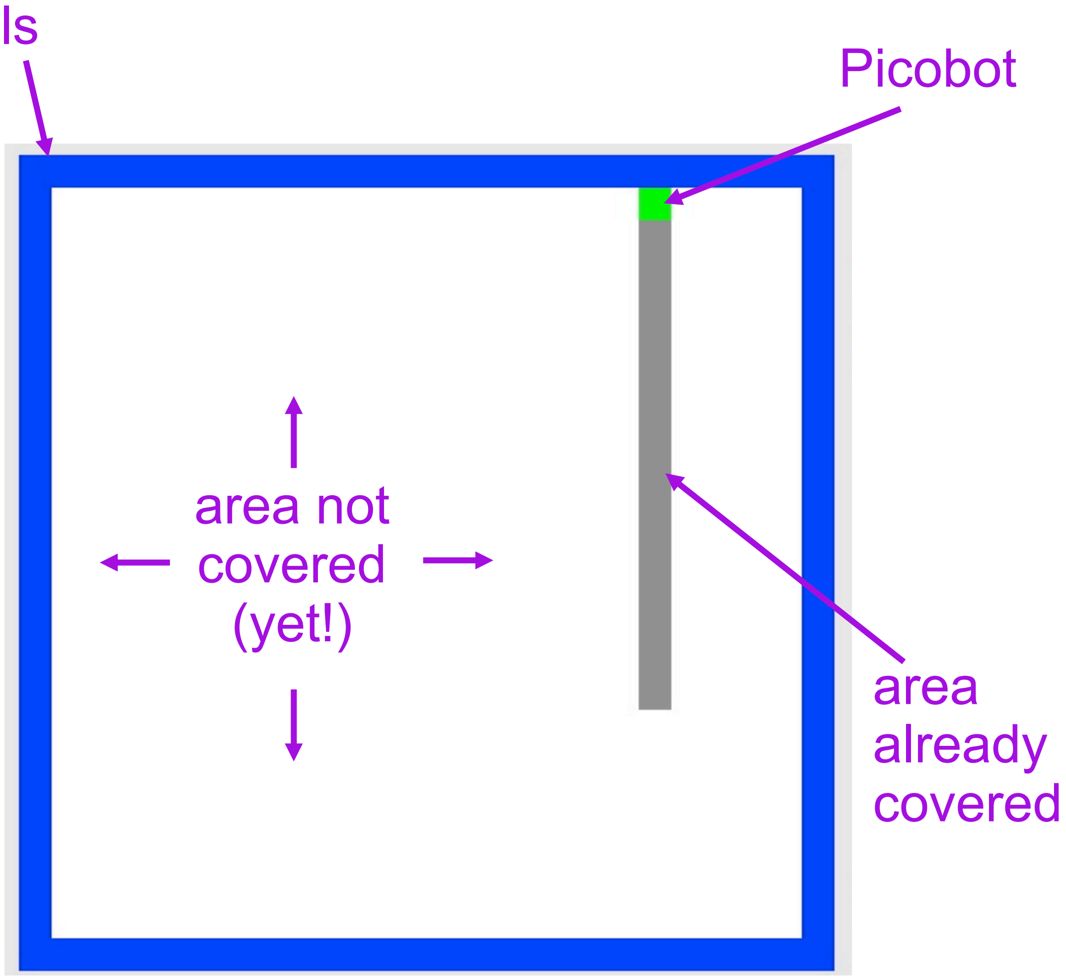
`www.cs.hmc.edu/picobot`

Introductions: Picobot



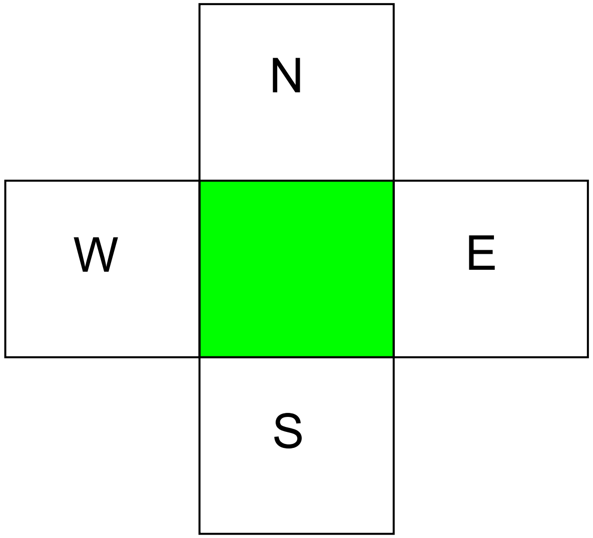
Murata Girl

Roomba

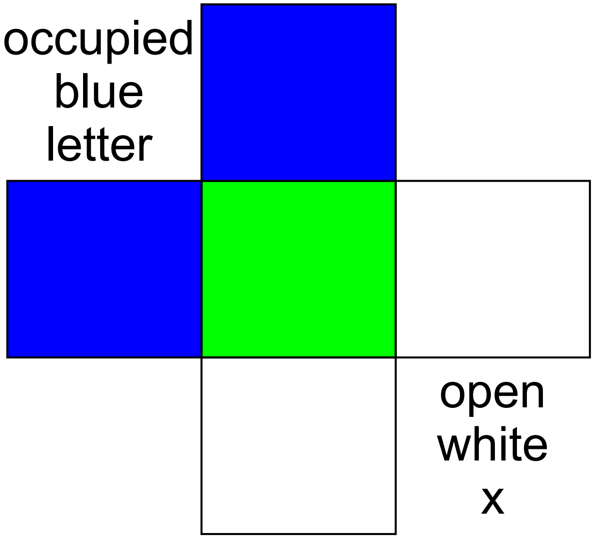


Goal: whole-environment coverage with only ***local sensing***…

Environment in the NEWS!



Picobot can only sense things directly to the N, E, W, and S



We can represent a particular environment with a text “code”.

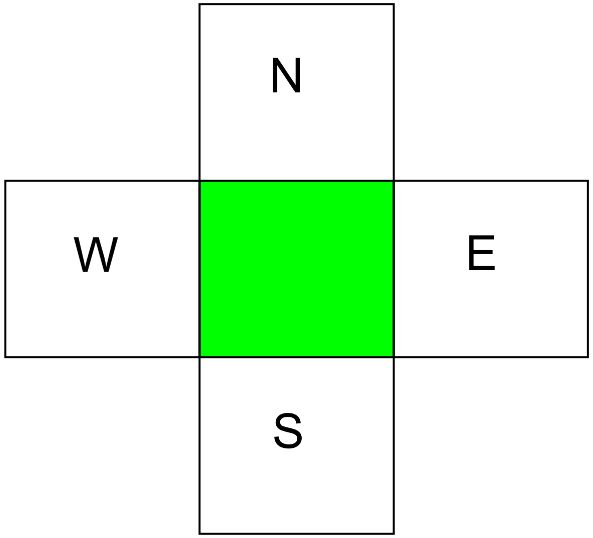
Nx**W**x

↑ ↑ ↑ ↑

N E W S

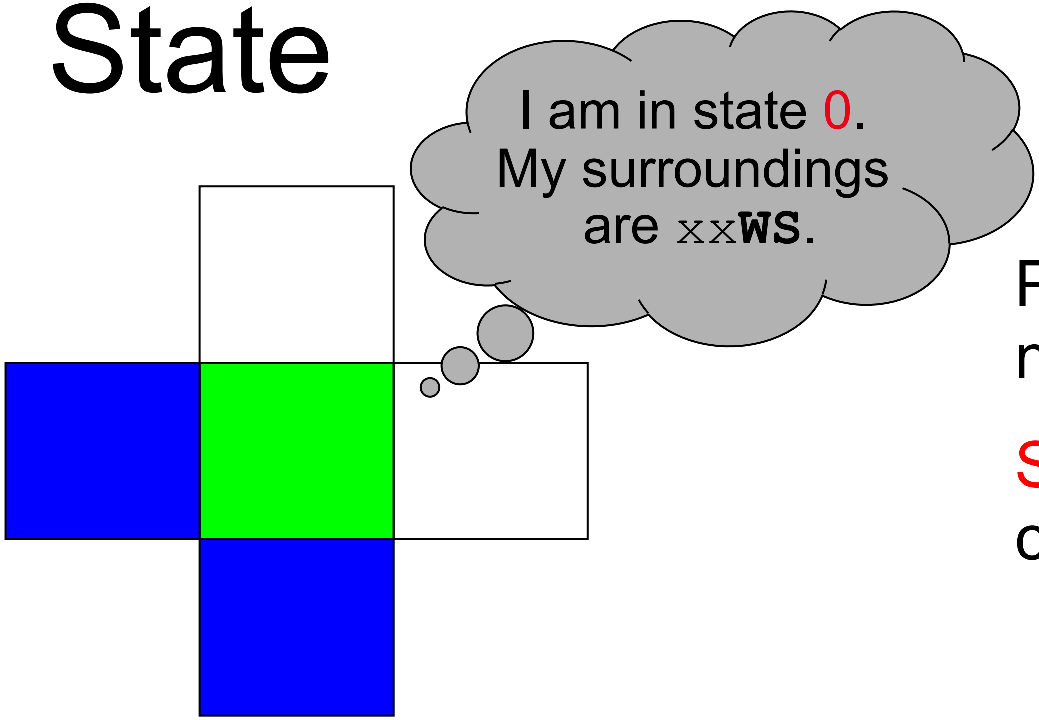
Surroundings are always in NEWS order.

Surroundings



How many distinct surroundings are there?

State



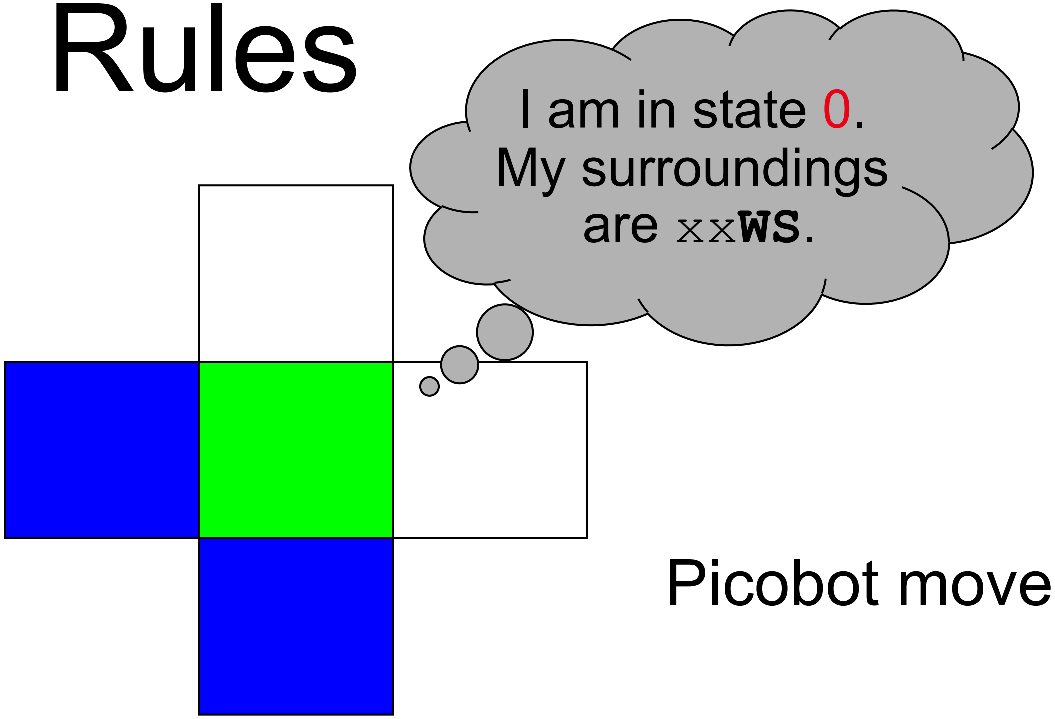
Picobot's memory is a single number, called its state.

State is the *internal context* of computation.

Picobot always starts in state 0.

State and **surroundings** represent everything the robot knows about the world

Rules



Aha!
I should move N.
I should enter state 0.

Picobot moves according to a set of rules:

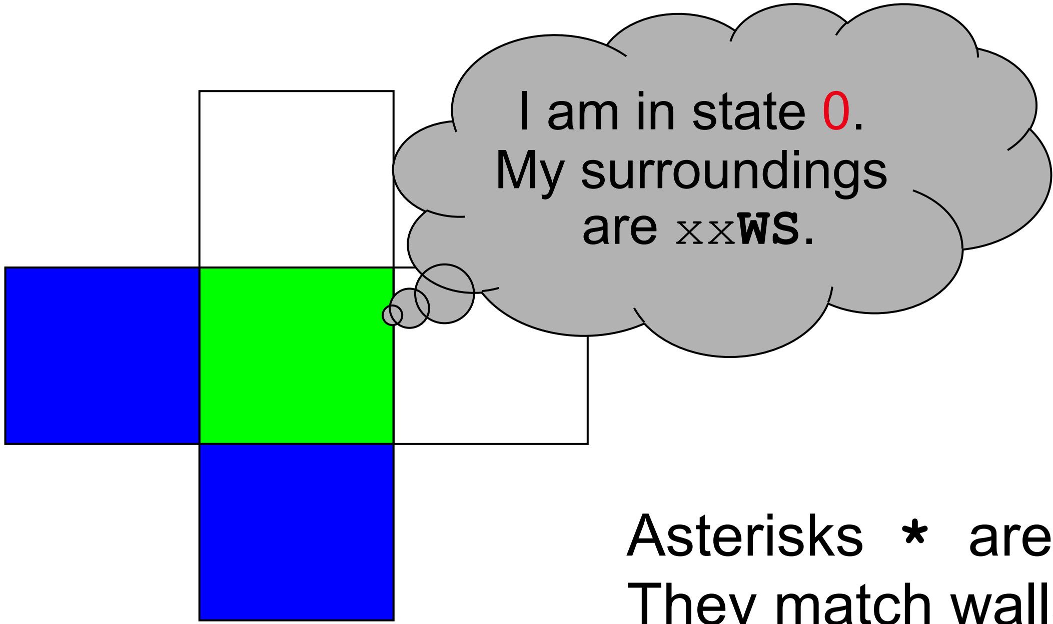
A capital “X” here means “Don’t Move”

state	surroundings		direction	new state
0	xx**WS**	→	N	0

If I'm in state 0
seeing xx**WS***,*

Then I move **N***orth, and*
change to state 0*.*

Wildcards



Aha! This matches x***

Asterisks * are wild cards.
They match walls ***or*** empty space:

state	surroundings		direction	new state
0	x***	→	N	0

and EWS may be wall or ***empty space***

N must be empty

Q

state	surroundings		direction	new state
0	`x***`	`->`	N	0
0	`N***`	`->`	X	0

A capital “X”
here means
“Don’t Move”

- Picobot checks its rules from the top each time.
- When it finds a matching rule, that rule runs.
- Only one rule is allowed per state and surroundings.

1. What will this set of rules do to Picobot?

2. Try to add some rules so that we go to the bottom now and then back up forever! (Hint: it will require adding a state 1)

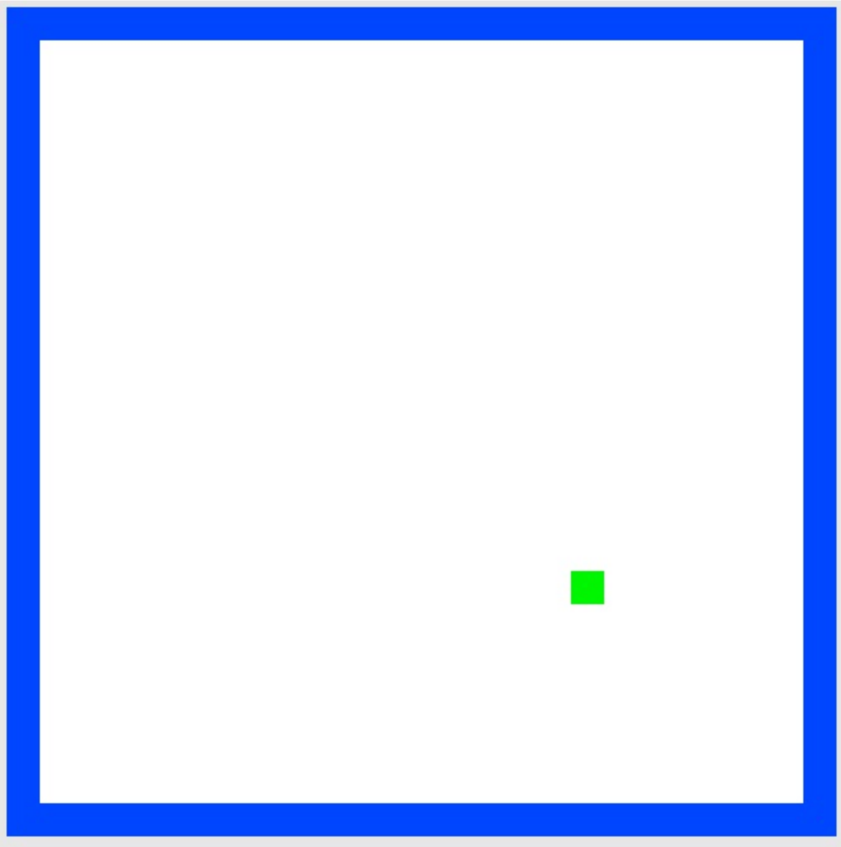
Worksheet

This week!

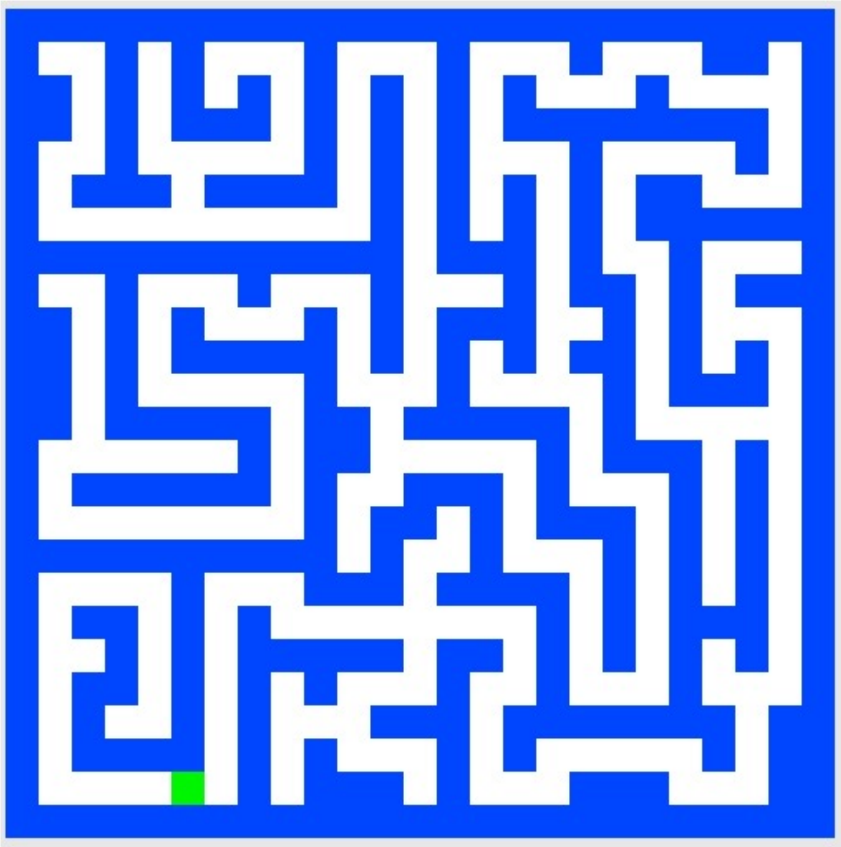
Write rules that will always cover these two rooms.

(*separate* sets of rules are encouraged…)

Lab Problem 2



Problem 4



Your “program” can be slow but it should work for any starting location and for any wall-connected maze!

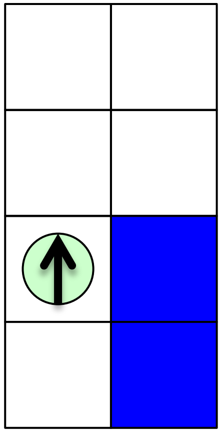
DEMO!

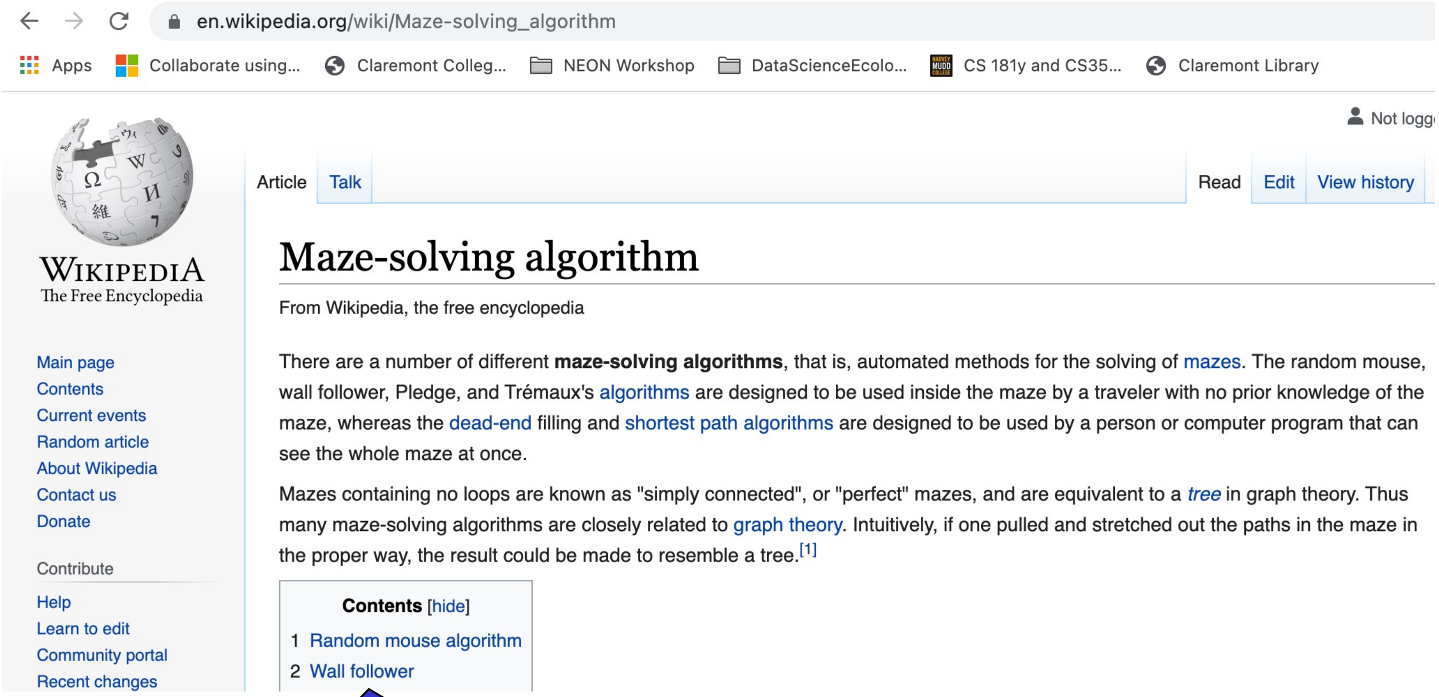
our best: 3 states, 7 rules

our best: 4 states, 8 rules

Hint: a word about states

Imagine state 0 means “pointing north”
state 1 means “pointing east”
state 2 …
state 3 …





Maze-solving algorithm

From Wikipedia, the free encyclopedia

There are a number of different **maze-solving algorithms**, that is, automated methods for the solving of mazes. The random mouse, wall follower, Pledge, and Trémaux's algorithms are designed to be used inside the maze by a traveler with no prior knowledge of the maze, whereas the dead-end filling and shortest path algorithms are designed to be used by a person or computer program that can see the whole maze at once.

Mazes containing no loops are known as "simply connected", or "perfect" mazes, and are equivalent to a *tree* in graph theory. Thus many maze-solving algorithms are closely related to graph theory. Intuitively, if one pulled and stretched out the paths in the maze in the proper way, the result could be made to resemble a tree.[1]

Contents [hide]

Problem 4: Right-hand rule

Introductions: Python

Richer syntax allows greater expressiveness!

```python
def alife_sim(num_gens, pop_size, num_to_select, network, inhibitor_l):
    """Do an artificial life simulation for numGens generations with
    popSize organisms."""

    # create initial population
    fit_d = {}
    pop_l = []
    for org in create_initial_pop(pop_size, network, inhibitor_l):
        fitness = org.get_fitness()
        pop_l.append((fitness, org))
        fit_d[hash(org)] = fitness

    # simulate
    top_l = get_top_orgs(pop_l, num_to_select)    # get top orgs
    for i in range(num_gens):
        pop_l = []
        for j in range(pop_size):
            to_replicate = random.choice(top_l)
            new_org = to_replicate[1].replicate()
            # get fitness
            if hash(new_org) in fit_d:
                fitness = fit_d[hash(new_org)]
            else:
                fitness = new_org.get_fitness()
                fit_d[hash(new_org)] = fitness
            pop_l.append((fitness, new_org))
        topL = get_top_orgs(pop_l, num_to_select)
        print("gen:", i, ":", top_l[0])
        if i%50 == 0:
            fit_d.clear()
    return top_l[0]
```

Learning to program is a bit like learning a foreign language!

Strange syntax!
Funky grammar

the ONION

Rules Grammar Change

English Traditional Replaced To Be New Syntax With

Why Python?

After all, there are thousands of languages to choose from!

- Relatively “nice” syntax
- Emerging as language of choice in many fields
- Packages for graphics, audio, scientific computing, …

```
print("Hello World!")
```

Python

R

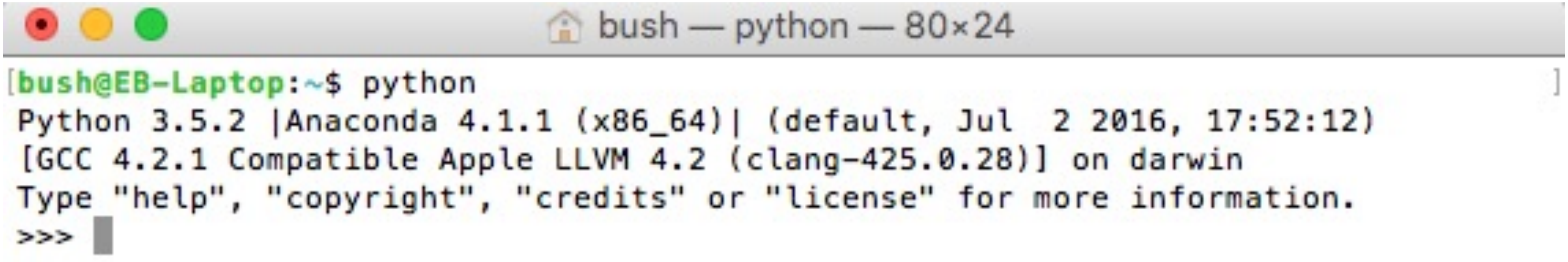
```
class HelloWorld {
  static public void main( String args[] ) {
    System.out.println( "Hello World!" );
  }
}
```

Java

```
#include <iostream.h>

main()
{
    cout << "Hello World!" << endl;
    return 0;
}
```

C++

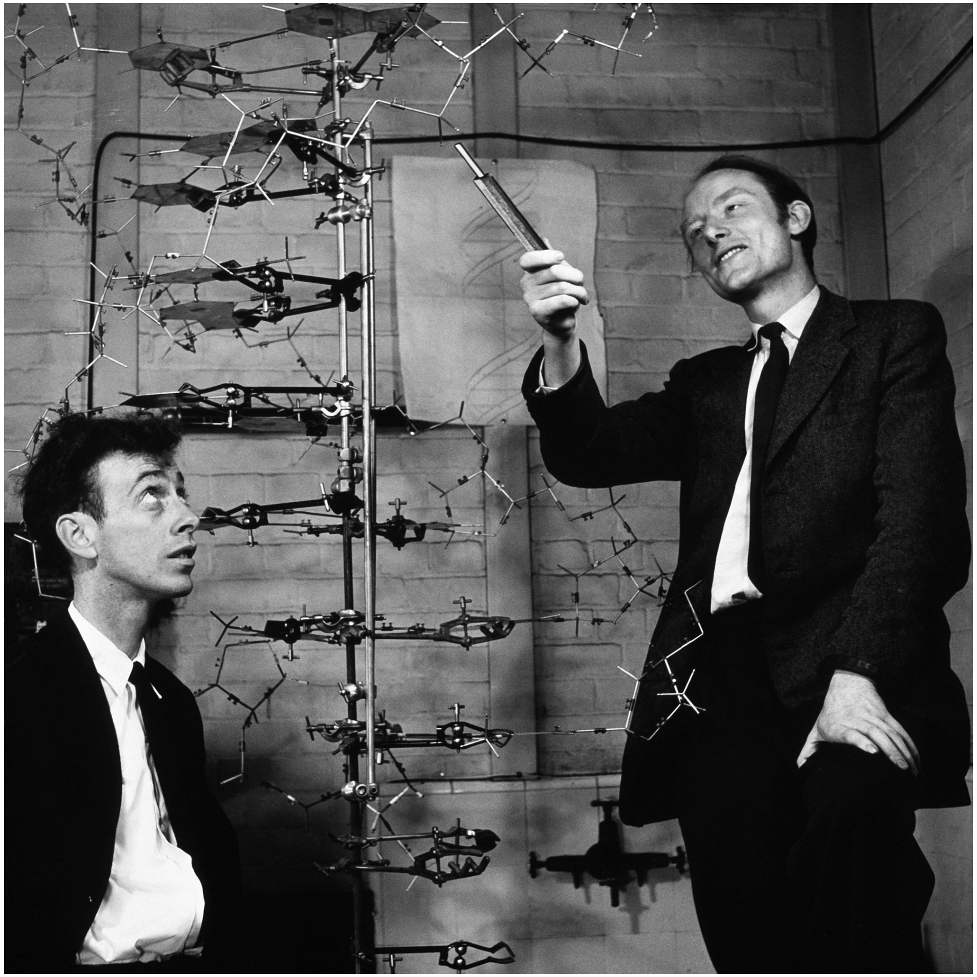
The Python interpreter



```
bush@EB-Laptop:~$ python
Python 3.5.2 |Anaconda 4.1.1 (x86_64)| (default, Jul  2 2016, 17:52:12)
[GCC 4.2.1 Compatible Apple LLVM 4.2 (clang-425.0.28)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

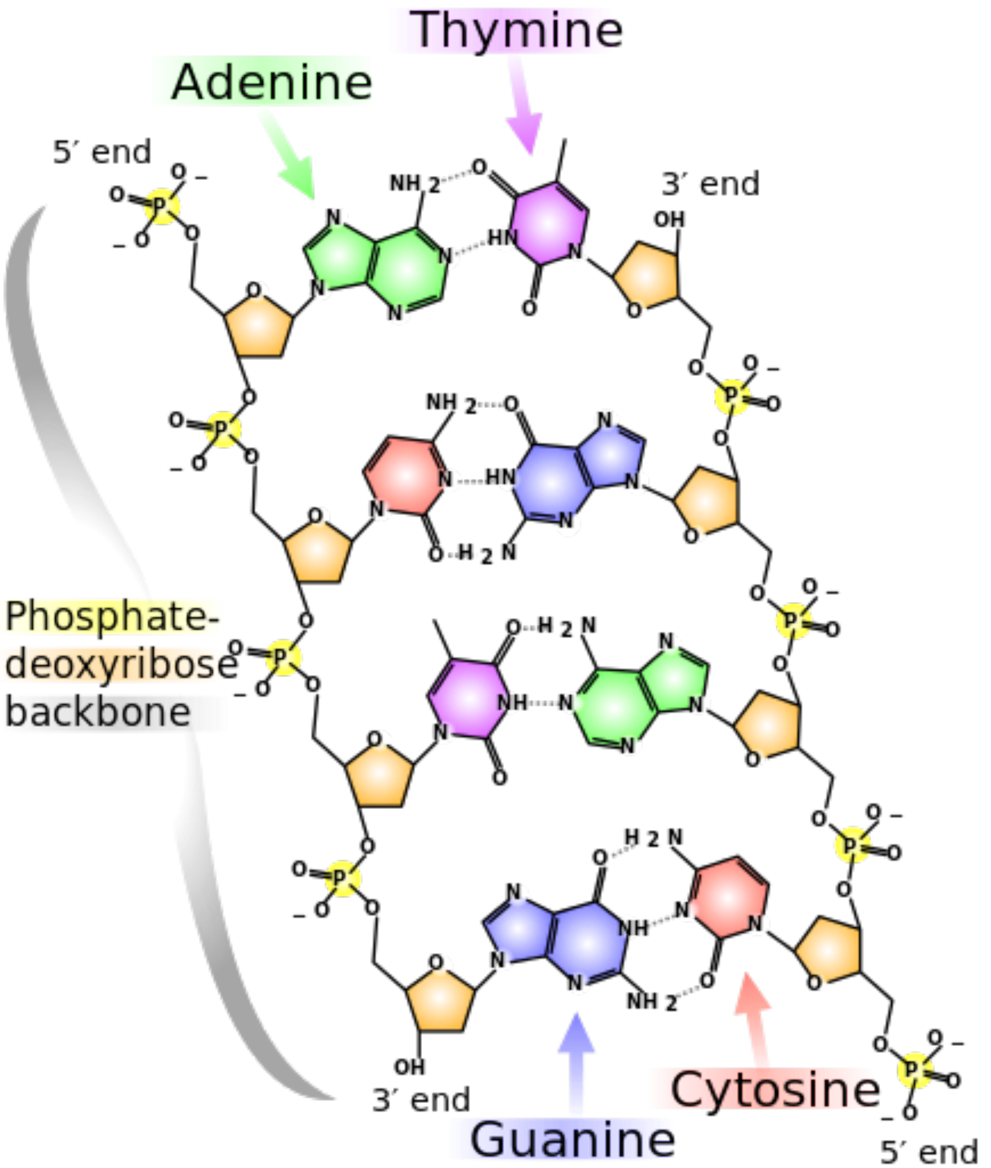
Python strings

```
>>> biologist1 = "Watson"
>>> biologist1
'Watson'



>>> biologist2 = 'Crick'
>>> biologist2
'Crick'
```



http://blogs.nature.com/freeassociation/tag/watson-and-crick

DNA is double stranded



http://upload.wikimedia.org/wikipedia/commons/thumb/e/e4/DNA_chemical_structure.svg/450px-DNA_chemical_structure.svg.png

Representing DNA molecules on a computer

```
5' - AATGCCGTGCTTGTAGACGTA - 3'
3' - TTACGGCACGAACATCTGCAT - 5'
```

By convention, we represent as a single string going 5' to 3’.

```
AATGCCGTGCTTGTAGACGTA

    or

TACGTCTTCAAGCACGGCATT
```

- Either of these two strings could be be used
- These are reverse complements of each other

Using strings: length and index

```
>>> myDNA = "AATGCCGTGCTT"

>>> len(myDNA)
12

>>> myDNA[0]
'A'

>>> myDNA[3]
'G'

>>> myDNA[20]
IndexError: string index out of range
```

0	1	2	3	4	5	6	7	8	9	10	11
A	A	T	G	C	C	G	T	G	C	T	T

Using strings: slicing

```
>>> myDNA = "AATGCCGTGCTT"

>>> myDNA[0:4]
'AATG'

>>> myDNA[3:7]
'GCCG'

>>> myDNA[1:]
'ATGCCGTGCTT'

>>> myDNA[:4]
'AATG'

>>> myDNA[10:42]
'TT'
```

0	1	2	3	4	5	6	7	8	9	10	11
A	A	T	G	C	C	G	T	G	C	T	T

Python Demo

Reminders:

- Introductory Survey (https://forms.gle/HMqDGHNjMHTfFHHL6)
- Lecture feedback form (https://forms.gle/aPmkpXDUTp4Xo4CV7)

Next lecture:

- More Python syntax to help analyze DNA sequences
- Fun with functions!

Names ______________________________________

state	surroundings		direction	new state
0	x***	->	N	0
0	N***	->	X	0

A capital “X” here means “Don’t Move”

- Picobot checks its rules from the top each time.
- When it finds a matching rule, that rule runs.
- Only one rule is allowed per state and surroundings.

1. What will this set of rules do to Picobot?

2. Try to add some rules so that we go to the bottom now and then back up forever! (Hint: it will require adding a state 1)

Worksheet