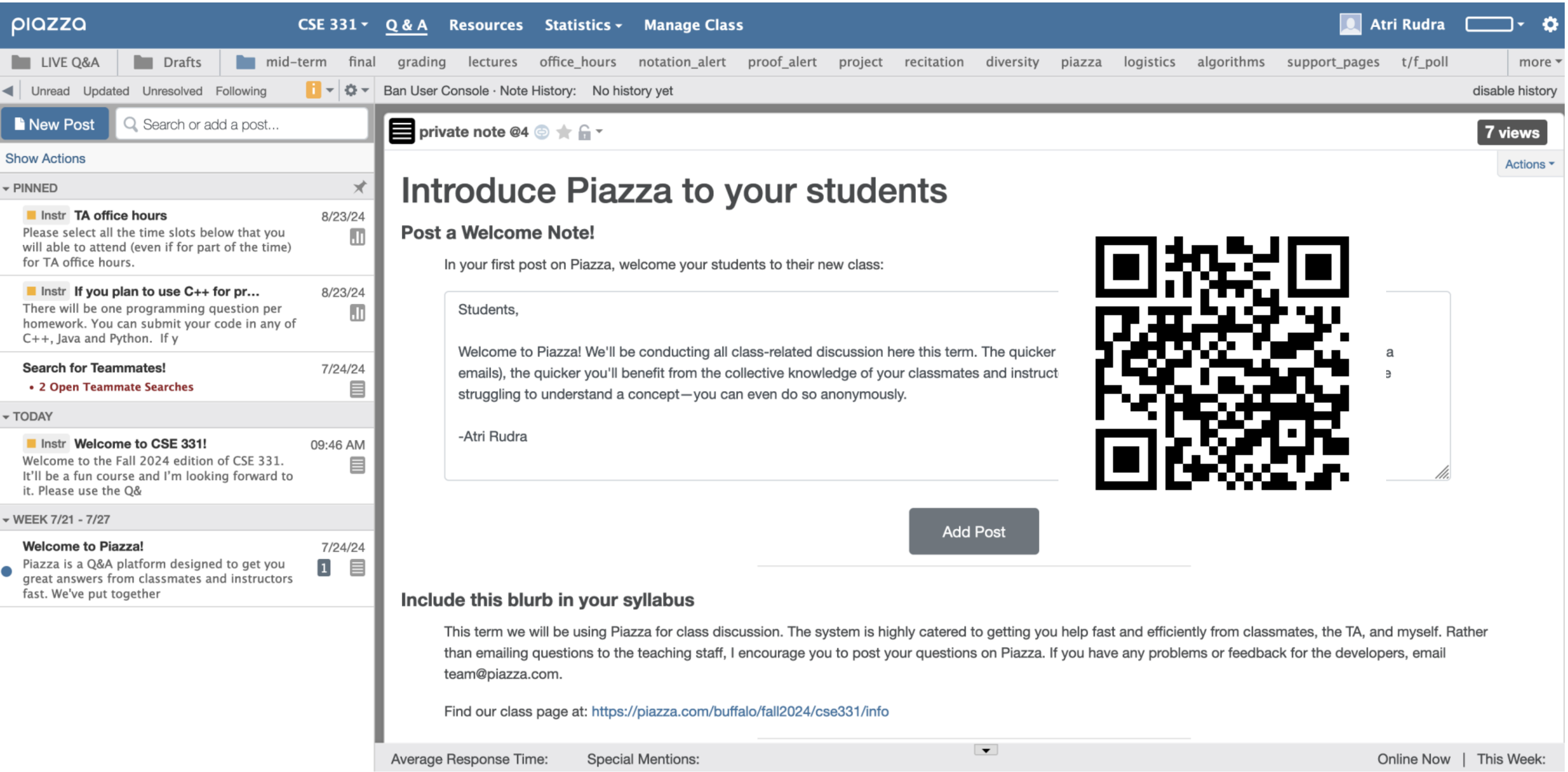
# Lecture 2

CSE 331

Aug 28, 2024

# Make sure you are on Piazza



Instr **TA office hours** 8/23/24
Please select all the time slots below that you will able to attend (even if for part of the time) for TA office hours.

Instr **If you plan to use C++ for pr...** 8/23/24
There will be one programming question per homework. You can submit your code in any of C++, Java and Python. If y

**Search for Teammates!** 7/24/24
• 2 Open Teammate Searches

Instr **Welcome to CSE 331!** 09:46 AM
Welcome to the Fall 2024 edition of CSE 331. It'll be a fun course and I'm looking forward to it. Please use the Q&

**Welcome to Piazza!** 7/24/24
Piazza is a Q&A platform designed to get you great answers from classmates and instructors fast. We've put together

## Introduce Piazza to your students

### Post a Welcome Note!

In your first post on Piazza, welcome your students to their new class:

Students,

Welcome to Piazza! We'll be conducting all class-related discussion here this term. The quicker emails), the quicker you'll benefit from the collective knowledge of your classmates and instruct struggling to understand a concept—you can even do so anonymously.

-Atri Rudra

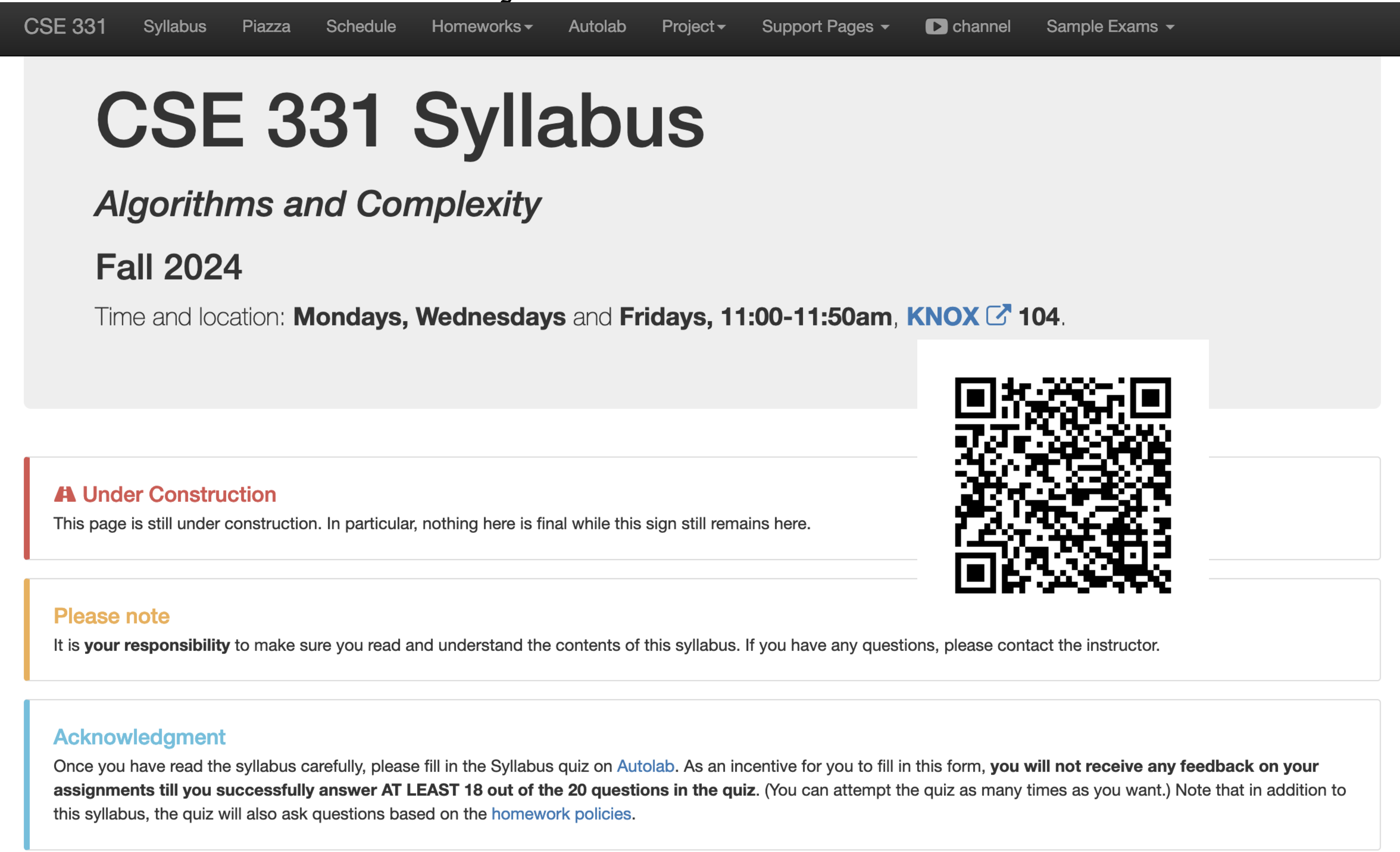
### Include this blurb in your syllabus

This term we will be using Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. If you have any problems or feedback for the developers, email team@piazza.com.

Find our class page at: https://piazza.com/buffalo/fall2024/cse331/info

https://piazza.com/buffalo/fall2024/cse331/

# Read the syllabus CAREFULLY!



## CSE 331 Syllabus

### *Algorithms and Complexity*

### Fall 2024

Time and location: **Mondays, Wednesdays** and **Fridays, 11:00-11:50am**, **KNOX** **104**.

**Under Construction**

This page is still under construction. In particular, nothing here is final while this sign still remains here.

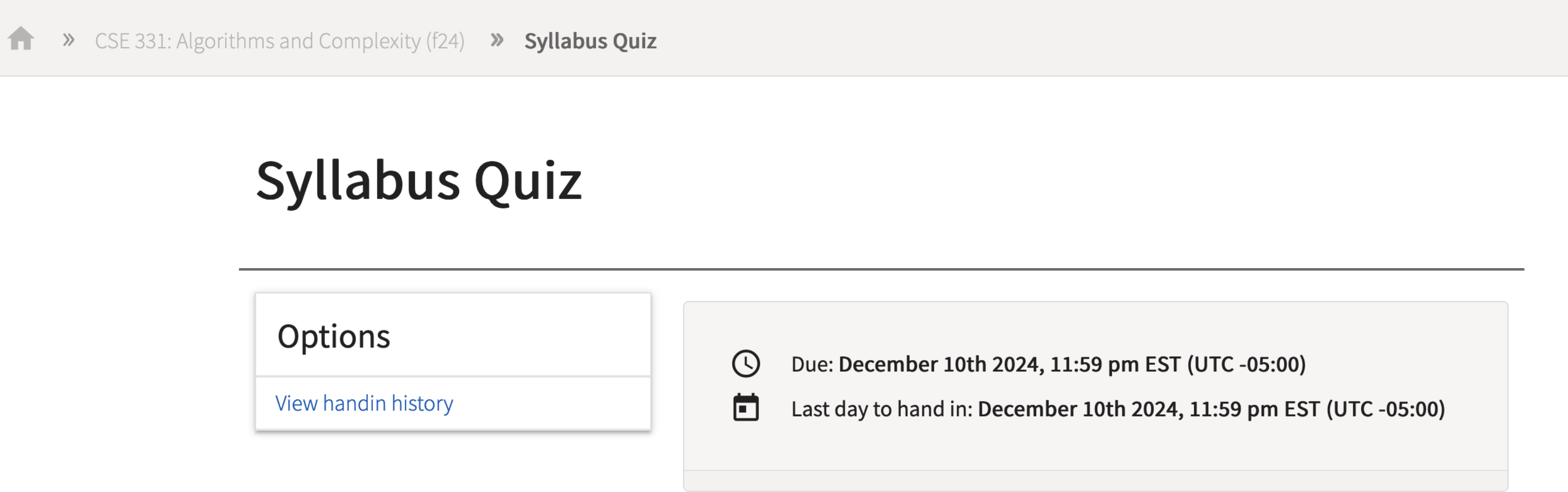
**Please note**

It is **your responsibility** to make sure you read and understand the contents of this syllabus. If you have any questions, please contact the instructor.

**Acknowledgment**

Once you have read the syllabus carefully, please fill in the Syllabus quiz on Autolab. As an incentive for you to fill in this form, **you will not receive any feedback on your assignments till you successfully answer AT LEAST 18 out of the 20 questions in the quiz**. (You can attempt the quiz as many times as you want.) Note that in addition to this syllabus, the quiz will also ask questions based on the homework policies.

# In spirit of trust but verify



## Syllabus Quiz

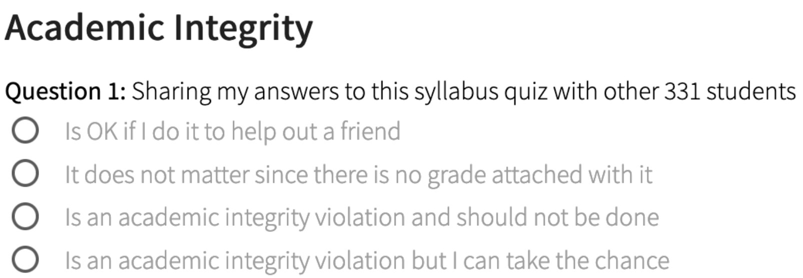
Options

View handin history

Due: **December 10th 2024, 11:59 pm EST (UTC -05:00)**

Last day to hand in: **December 10th 2024, 11:59 pm EST (UTC -05:00)**

No graded material will be handed back until you pass the syllabus quiz!



### Academic Integrity

**Question 1:** Sharing my answers to this syllabus quiz with other 331 students

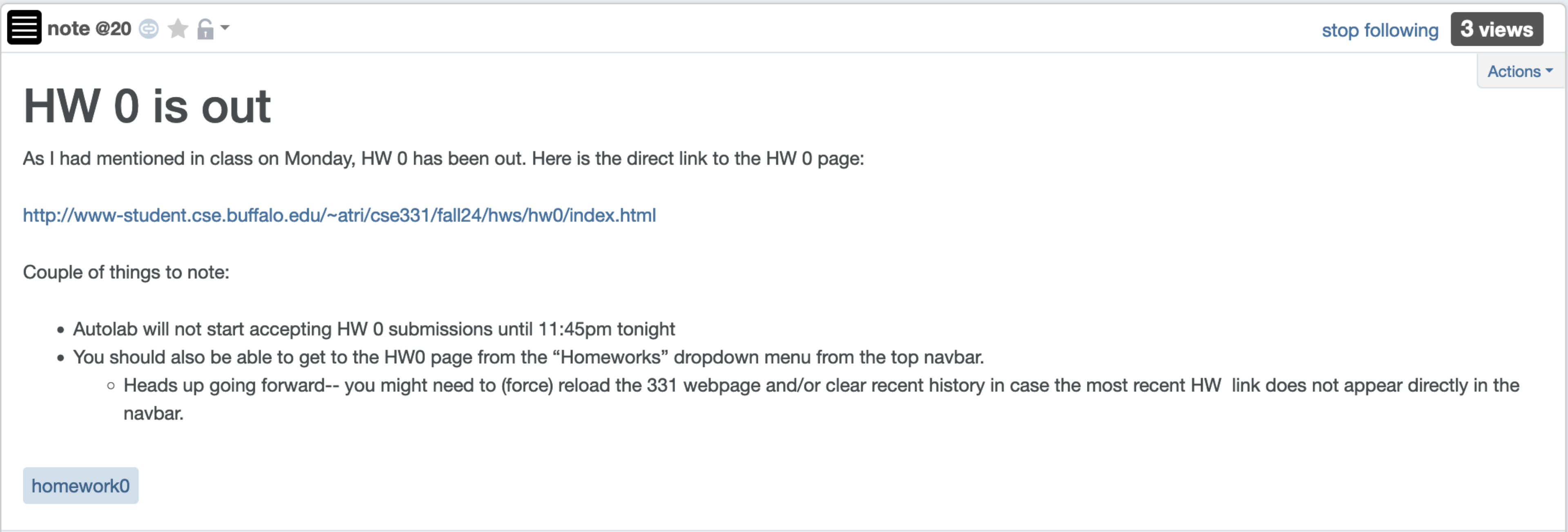
○ Is OK if I do it to help out a friend

○ It does not matter since there is no grade attached with it

○ Is an academic integrity violation and should not be done

○ Is an academic integrity violation but I can take the chance

# Homework 0 is out



## HW 0 is out

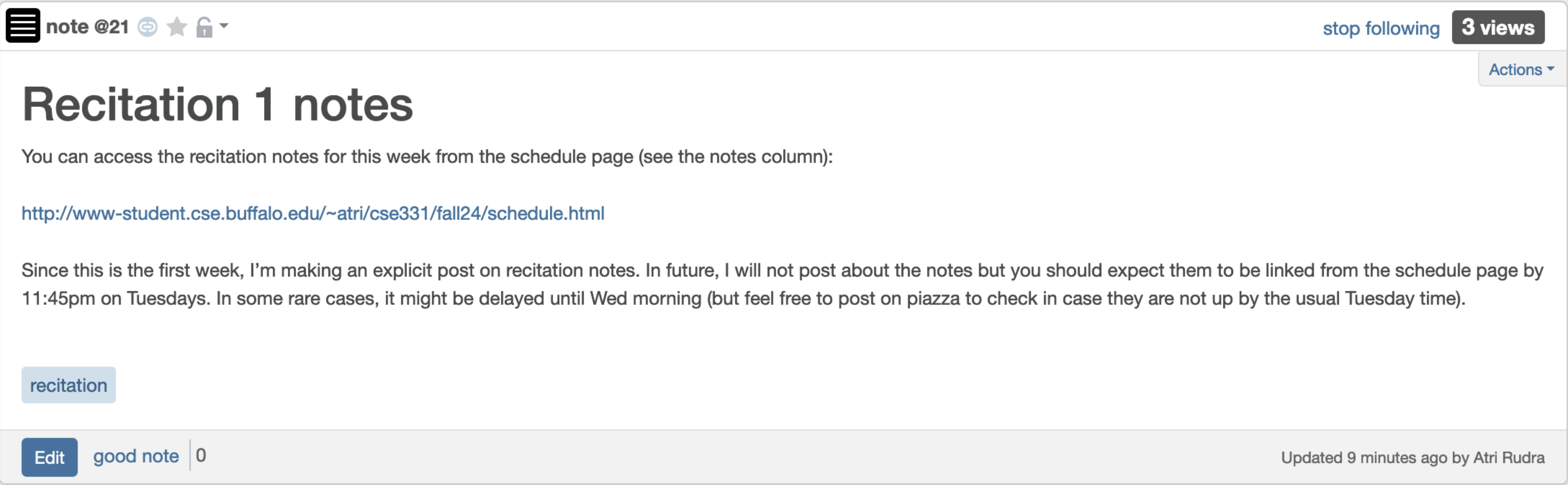
As I had mentioned in class on Monday, HW 0 has been out. Here is the direct link to the HW 0 page:

http://www-student.cse.buffalo.edu/~atri/cse331/fall24/hws/hw0/index.html

Couple of things to note:

- Autolab will not start accepting HW 0 submissions until 11:45pm tonight
- You should also be able to get to the HW0 page from the “Homeworks” dropdown menu from the top navbar.
  - Heads up going forward-- you might need to (force) reload the 331 webpage and/or clear recent history in case the most recent HW  link does not appear directly in the navbar.

# Recitation notes for this week



## Recitation 1 notes

You can access the recitation notes for this week from the schedule page (see the notes column):

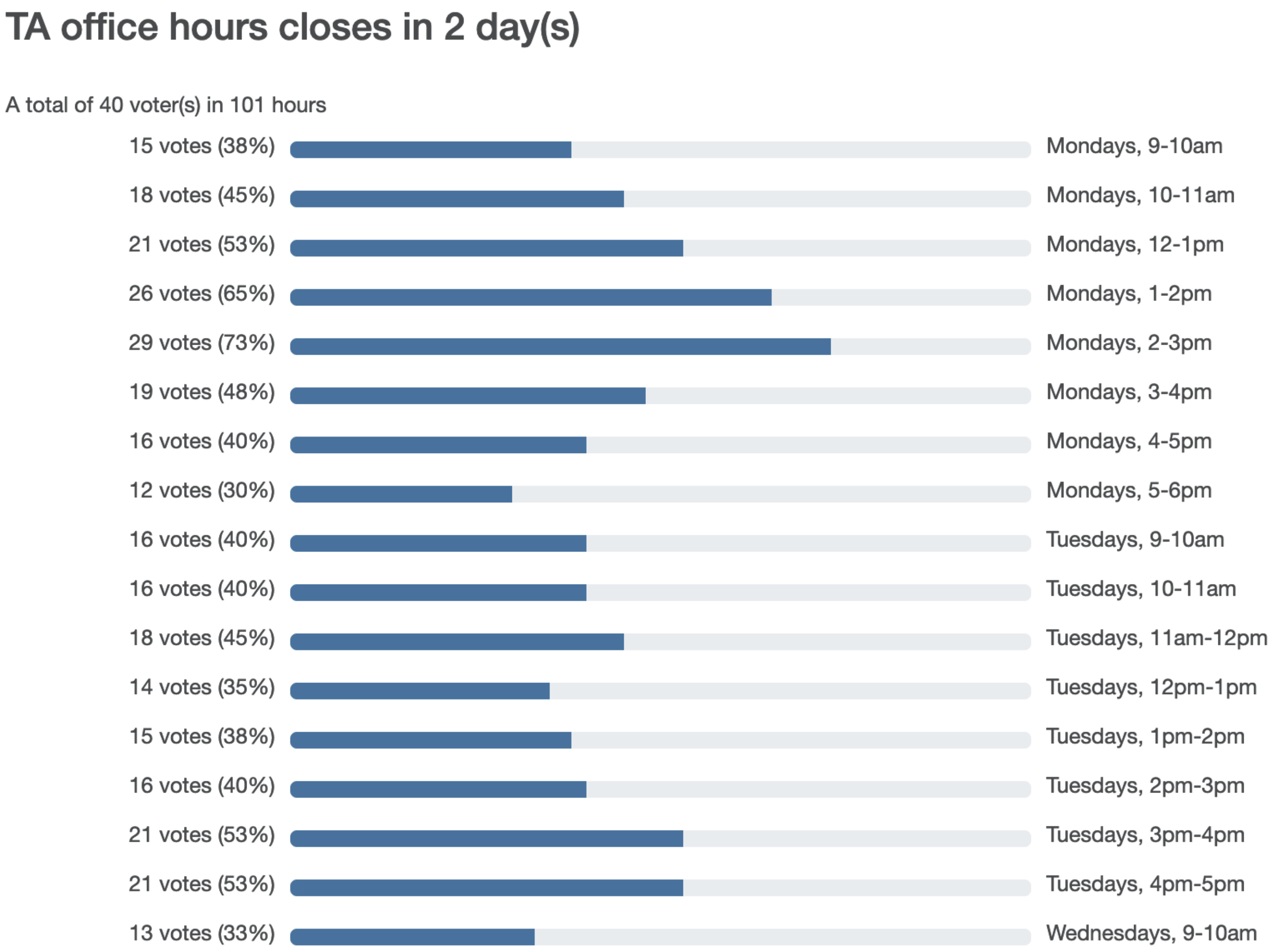
http://www-student.cse.buffalo.edu/~atri/cse331/fall24/schedule.html

Since this is the first week, I'm making an explicit post on recitation notes. In future, I will not post about the notes but you should expect them to be linked from the schedule page by 11:45pm on Tuesdays. In some rare cases, it might be delayed until Wed morning (but feel free to post on piazza to check in case they are not up by the usual Tuesday time).

Updated 9 minutes ago by Atri Rudra

Recitations start this week!

# TA Office hours finalized tomorrow

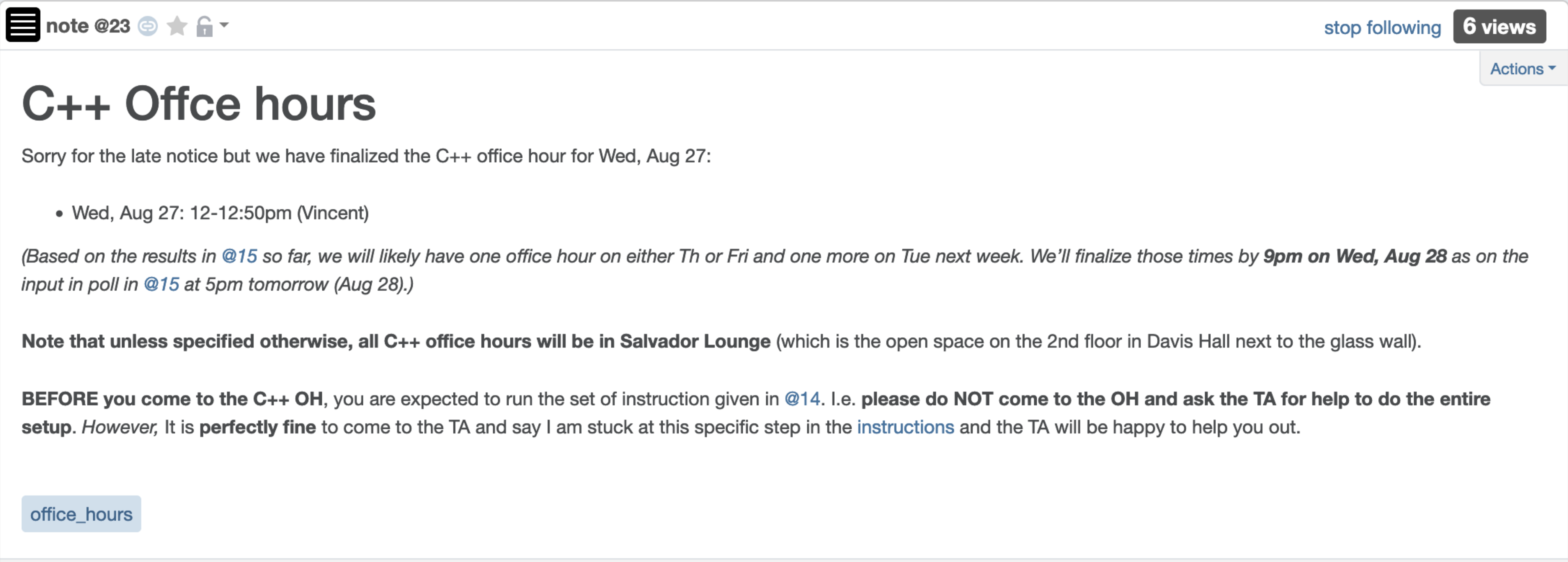


## TA office hours closes in 2 day(s)

A total of 40 voter(s) in 101 hours

| | |
|---|---|
| 15 votes (38%) | Mondays, 9-10am |
| 18 votes (45%) | Mondays, 10-11am |
| 21 votes (53%) | Mondays, 12-1pm |
| 26 votes (65%) | Mondays, 1-2pm |
| 29 votes (73%) | Mondays, 2-3pm |
| 19 votes (48%) | Mondays, 3-4pm |
| 16 votes (40%) | Mondays, 4-5pm |
| 12 votes (30%) | Mondays, 5-6pm |
| 16 votes (40%) | Tuesdays, 9-10am |
| 16 votes (40%) | Tuesdays, 10-11am |
| 18 votes (45%) | Tuesdays, 11am-12pm |
| 14 votes (35%) | Tuesdays, 12pm-1pm |
| 15 votes (38%) | Tuesdays, 1pm-2pm |
| 16 votes (40%) | Tuesdays, 2pm-3pm |
| 21 votes (53%) | Tuesdays, 3pm-4pm |
| 21 votes (53%) | Tuesdays, 4pm-5pm |
| 13 votes (33%) | Wednesdays, 9-10am |

# C++ OH times



## C++ Offce hours

Sorry for the late notice but we have finalized the C++ office hour for Wed, Aug 27:

- Wed, Aug 27: 12-12:50pm (Vincent)

*(Based on the results in @15 so far, we will likely have one office hour on either Th or Fri and one more on Tue next week. We'll finalize those times by **9pm on Wed, Aug 28** as on the input in poll in @15 at 5pm tomorrow (Aug 28).)*

**Note that unless specified otherwise, all C++ office hours will be in Salvador Lounge** (which is the open space on the 2nd floor in Davis Hall next to the glass wall).

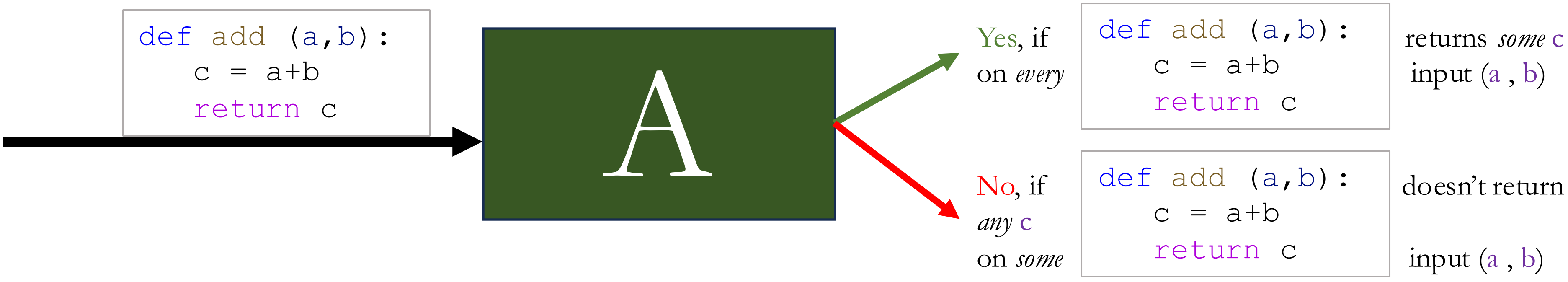
**BEFORE you come to the C++ OH**, you are expected to run the set of instruction given in @14. I.e. **please do NOT come to the OH and ask the TA for help to do the entire setup**. *However,* It is **perfectly fine** to come to the TA and say I am stuck at this specific step in the instructions and the TA will be happy to help you out.

# Mon recap: Halting Problem

*Input:* A program P

*Output:* Yes if P terminates on all possible inputs
No otherwise

Let A be a program that solves the Halting problem on all inputs



```
def add (a,b):
    c = a+b
    return c
```

Yes, if on *every*

```
def add (a,b):
    c = a+b
    return c
```

returns *some* c input (a , b)

No, if *any* c on *some*

```
def add (a,b):
    c = a+b
    return c
```

doesn’t return input (a , b)

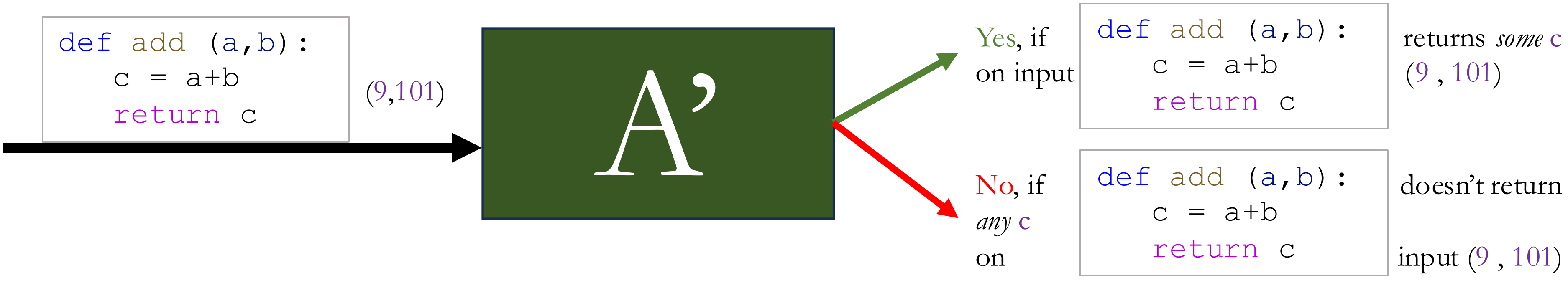
# Questions/Comments?

# Meta Q: Halting Problem (ver 2)

*Input:* A program P and input I

*Output:* Yes if P terminates on I
No otherwise

Let A’ be a program that solves the Halting problem (ver 2)



```
def add (a,b):
   c = a+b
   return c
```

(9,101)

A’

Yes, if on input

```
def add (a,b):
   c = a+b
   return c
```

returns *some* c (9 , 101)

No, if *any* c on

```
def add (a,b):
   c = a+b
   return c
```

doesn’t return input (9 , 101)

# Halting Problem: ver 1 vs. ver 2

*Input:* A program P

*Output:* Yes if P terminates on all possible inputs
No otherwise

*Input:* A program P and input I

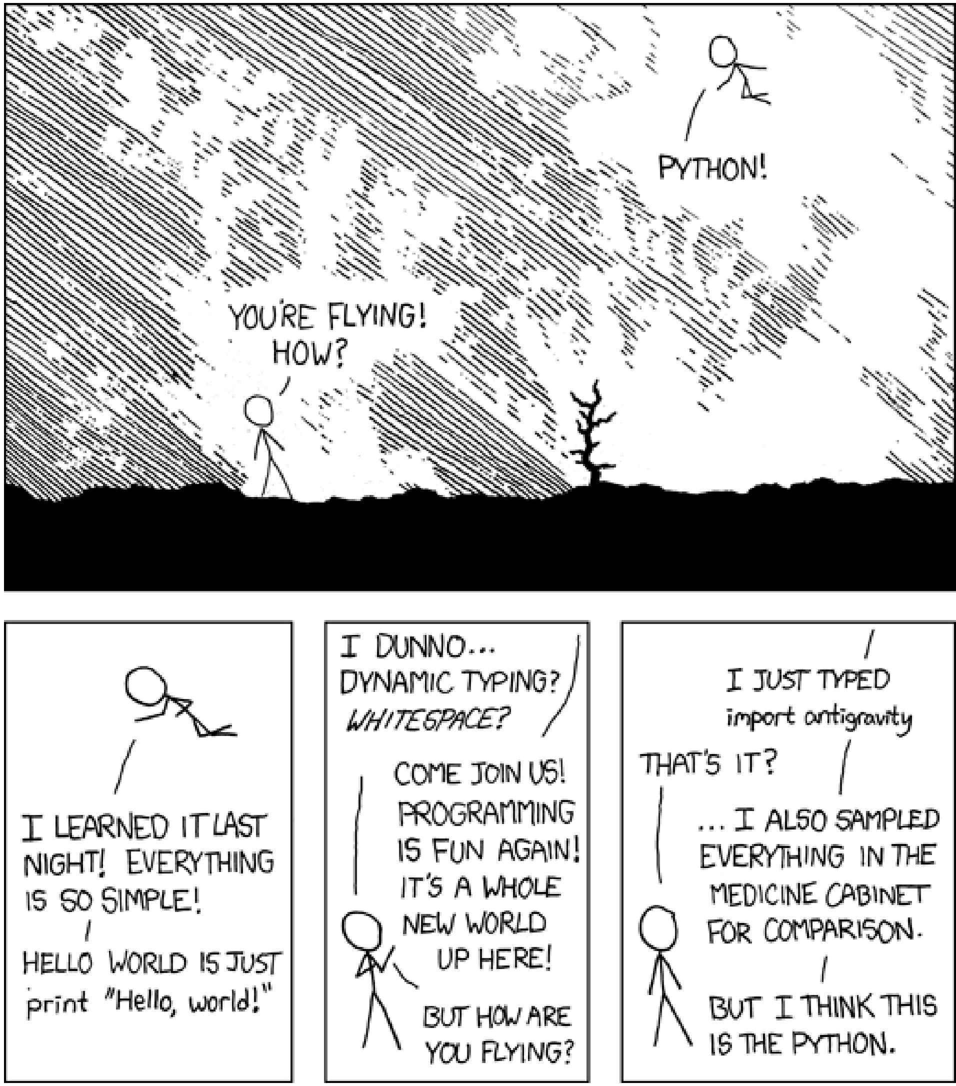
*Output:* Yes if P terminates on I
No otherwise

Ver 1 is harder than ver 2 (can *prove* this!)

Theorem: There is no magic box A' for ver 2 (as long as A' terminates)

# Notebook detour

Let’s define some functions in Python


PYTHON!
YOU'RE FLYING! HOW?
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?
I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

https://xkcd.com/353/

# Questions?

# Next up:

*Input:* A program P and input I

*Output:* Yes if P terminates on I
No otherwise

Theorem: There is no magic box A' for ver 2 (as long as A' terminates)

# Let us assume a magic_box for A’ exists!

```
def magic_box ( P, I):
# This is a magic box so there is no real code here!
''' Return True if P halts on I and False otherwise'''
```

Assume that

1. magic_box terminates on all inputs
2. magic_box ALWAYS correctly decides if P halts on I or not

# A new function contradiction

```
def contradiction ( P ): # This function takes a program as an input

#Run magic_box on (P,P)
if magic_box (P,P): # Use an UTM to make this call
   while True:
      pass # Do nothing

return # Just terminate if magic_box(P,P) returns False
```

Since we assumed magic_box exists contradiction is well defined!

# A function call

```
contradiction (contradiction) # Use an UTM to make this call
```

Since we assumed `contradiction` is well defined, the above is a legit function call!

# Wait, what?!!!

# `contradiction (contradiction)`

What are outcomes of the function call `contradiction (contradiction)`?

It terminates

It does not terminate

~~[illegible]~~

# Case 1:

`contradiction (contradiction)` terminates

```
def contradiction ( P ): # This function takes a program as an input

#Run magic_box on (P,P)
if magic_box (P,P): # Use an UTM to make this call
   while True:
      pass # Do nothing

return # Just terminate if magic_box(P,P) returns False
```

This does NOT terminate!!

```
contradiction (contradiction ):

if magic_box (contradiction, contradiction):
   while True:
      pass # Do nothing

return
```

What should this call return?

We get into an infinite loop here!

# Case 2:

`contradiction (contradiction)` does not terminate

```
def contradiction ( P ): # This function takes a program as an input

#Run magic_box on (P,P)
if magic_box (P,P): # Use an UTM to make this call
   while True:
      pass # Do nothing

return # Just terminate if magic_box(P,P) returns False
```

This DOES terminate!!

```
contradiction (contradiction ):

if magic_box (contradiction, contradiction):
   while True:
      pass # Do nothing

return
```

What should this call return?

We return here!

# Questions?

# Let’s recap: Argue `magic_box` doesn’t exist

This clearly is absurd!!

1. Assume `magic_box` exists

2. Defined a function `contradiction` that uses `magic_box`

3. Looked at the ONLY two possibilities

3.1. `contradiction(contradiction)` terminates

→ `contradiction(contradiction)` does NOT terminate

3.2. `contradiction(contradiction)` does NOT terminate

→ `contradiction(contradiction)` terminates

Which specific step in the above “argument” is wrong?

Congrats: You just did your 1st 331 proof!

# Questions?

# Proof Idea on the board…

$\frac{r}{k} \leq D^{R}(f) \leq D^{1}(f) \leq r \log(\frac{n}{r})$
$x_1$
$x_2$
$x_3$
$x_4$
$x_5$