CSE 331: Algorithms & Complexity

"A Proof"

Prof. Charlie Anne Carlson (She/Her)

Lecture 1

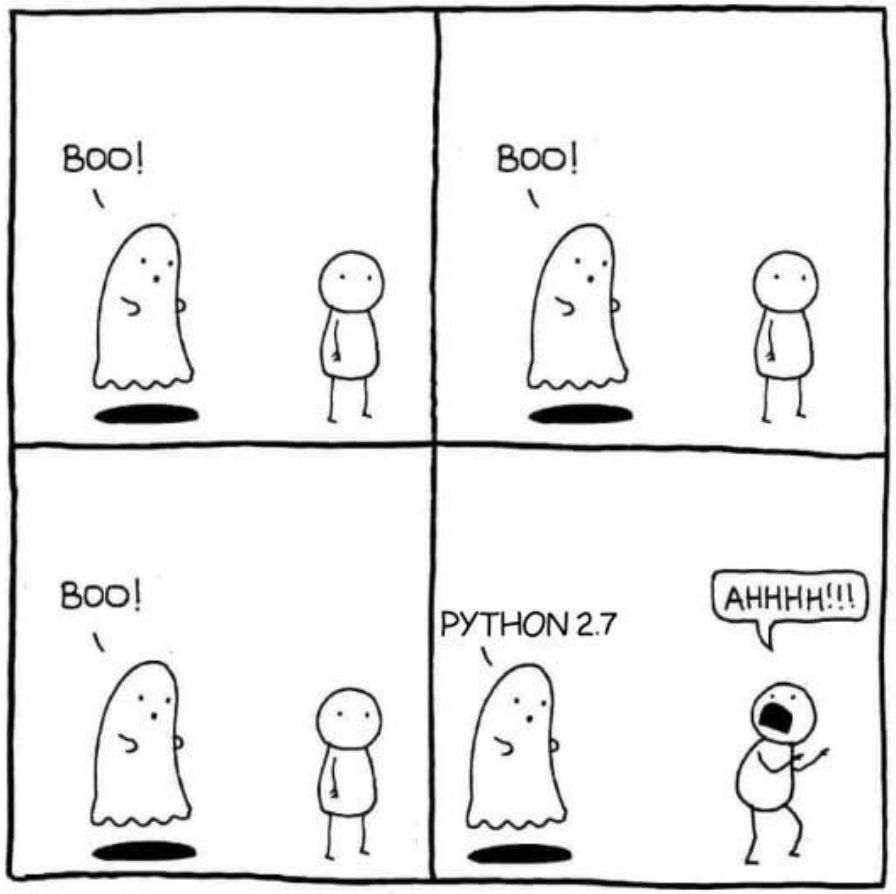
Monday August 27th, 2025

University at Buffalo®

Schedule

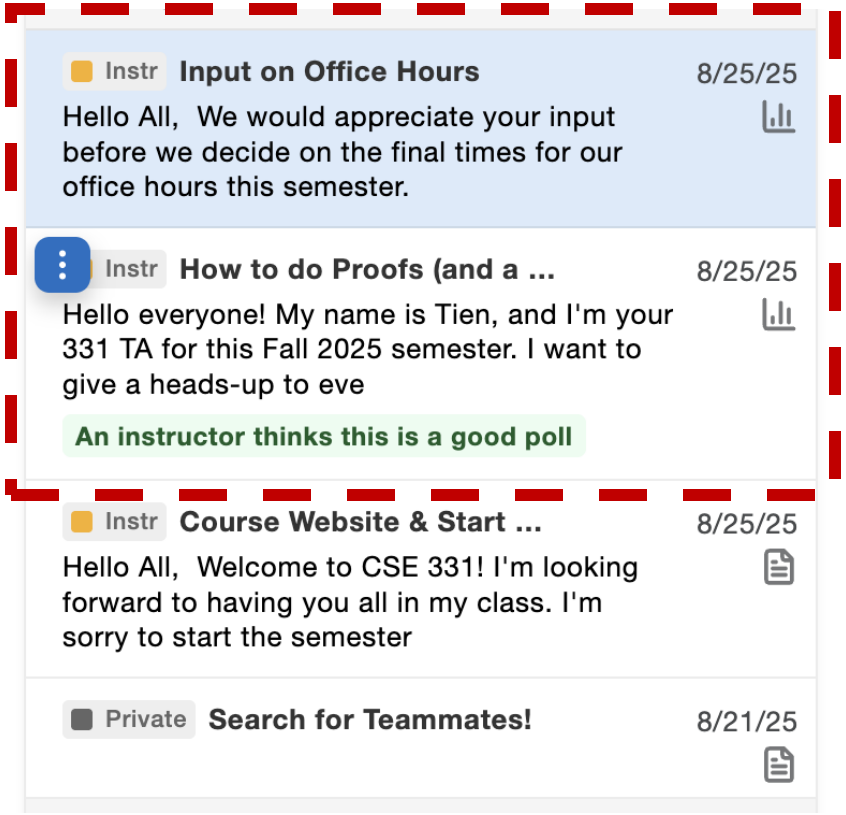
1. Course Updates
2. Advice
3. Halting Problem

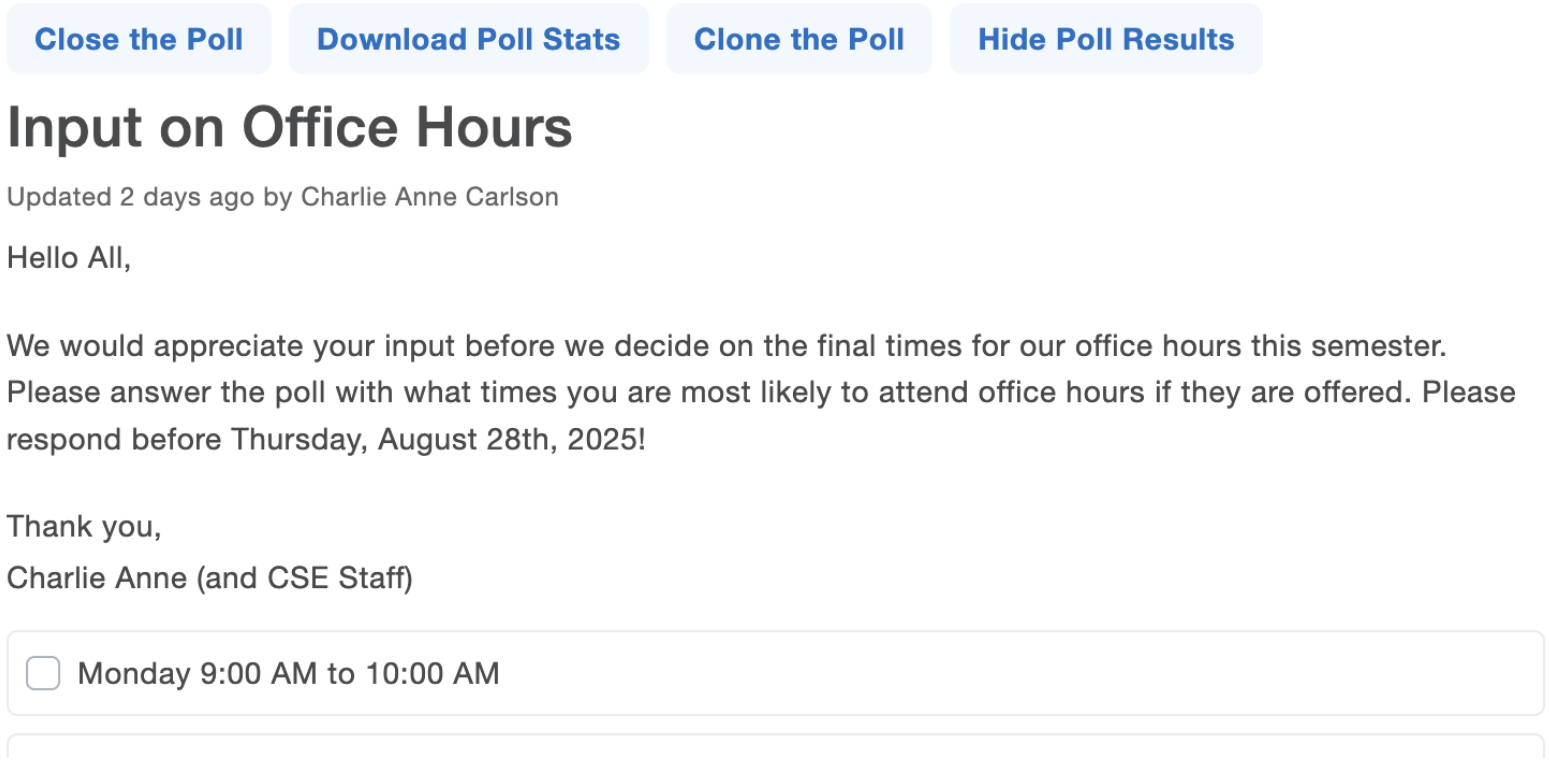
Course Updates: Syllabus Quiz Fixed!



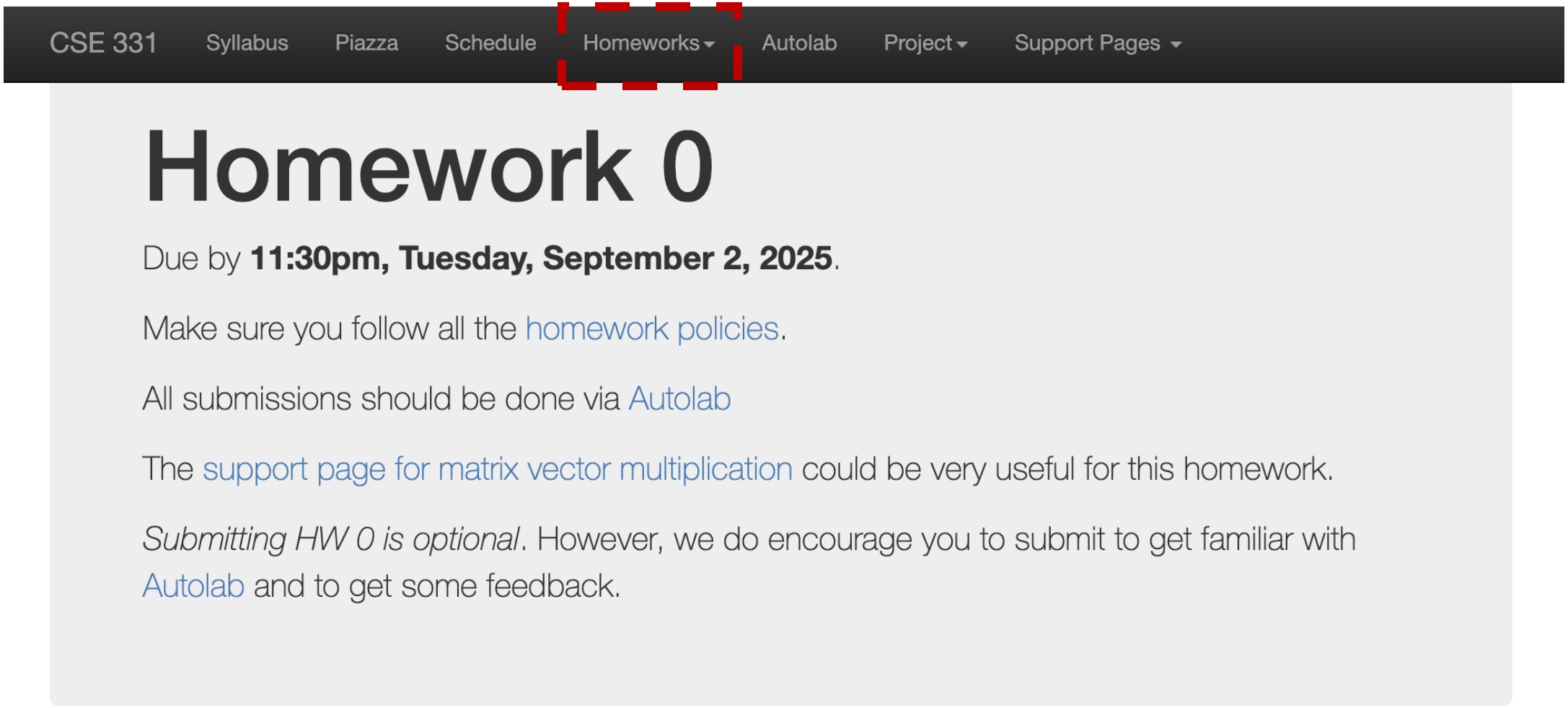
MrSavagePotato. “**Python 2.7.” Reddit, July 25, 2018.**
https://www.reddit.com/r/ProgrammerHumor/comments/91vtas/python_27/

Course Updates: Piazza Polls

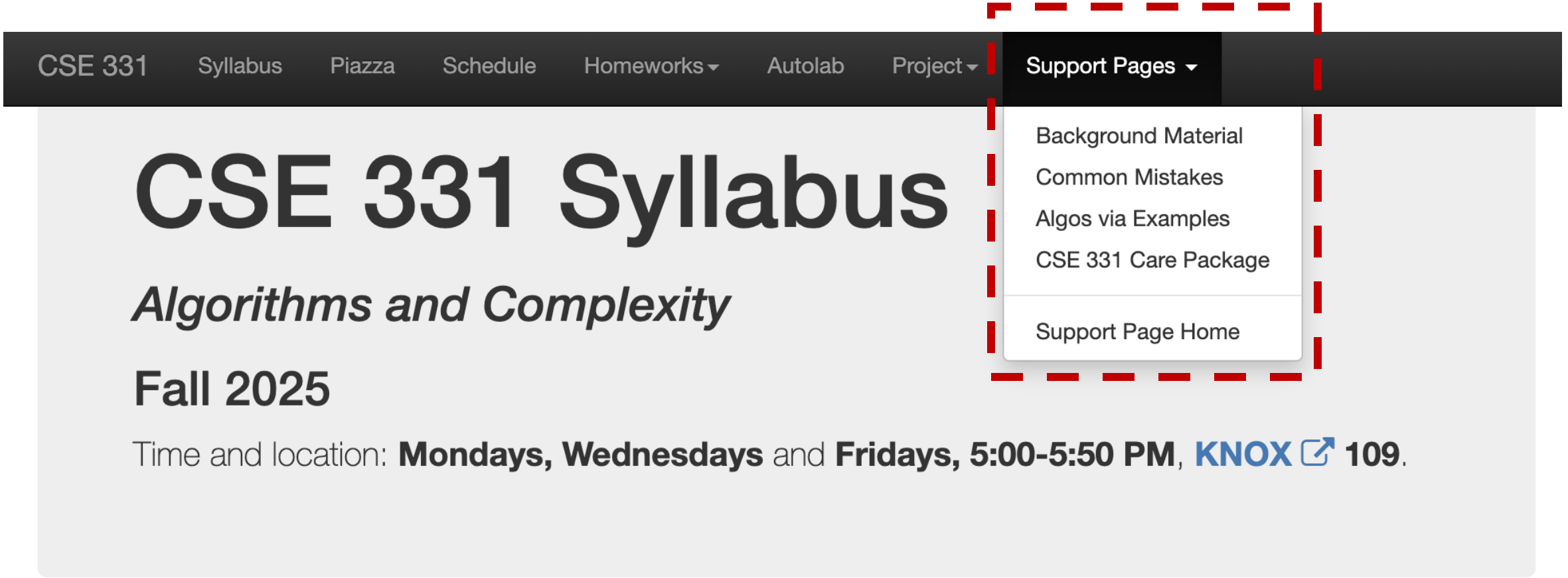




Course Updates: HW0 is out

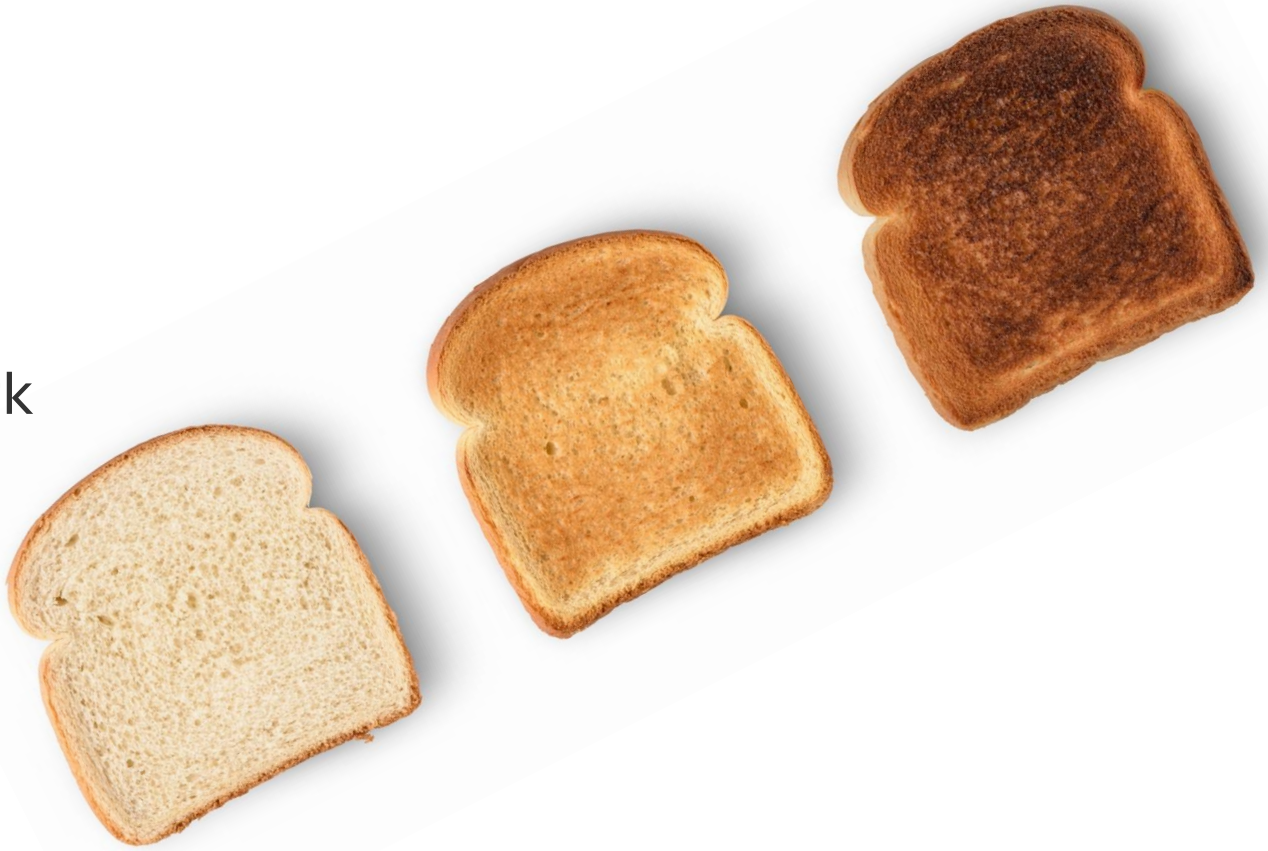


Advice: Support Materials



Advice: Solving Problems

1. Read problem carefully!
2. Think about problem for a few minutes.
3. Keep notes about ideas.
4. Step away when stuck.
5. Schedule time in advance to work on problems.
6. Go to office hours if stuck.
7. Turn in what you have before the deadline.



Advice: Big Three

1. Read Carefully
2. Don’t Cheat
3. Work Hard

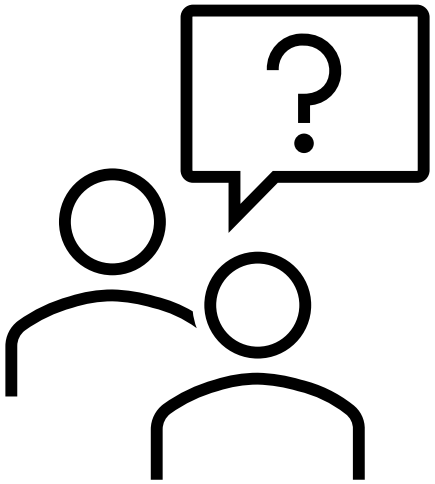
Questions?

Does it halt?

Q1: What does it mean for a program to "terminate?"

Q2: Can you think of program which does not terminate on any input?

Q3: Can you think of a program that terminates on all but one specific input?



Does it halt?

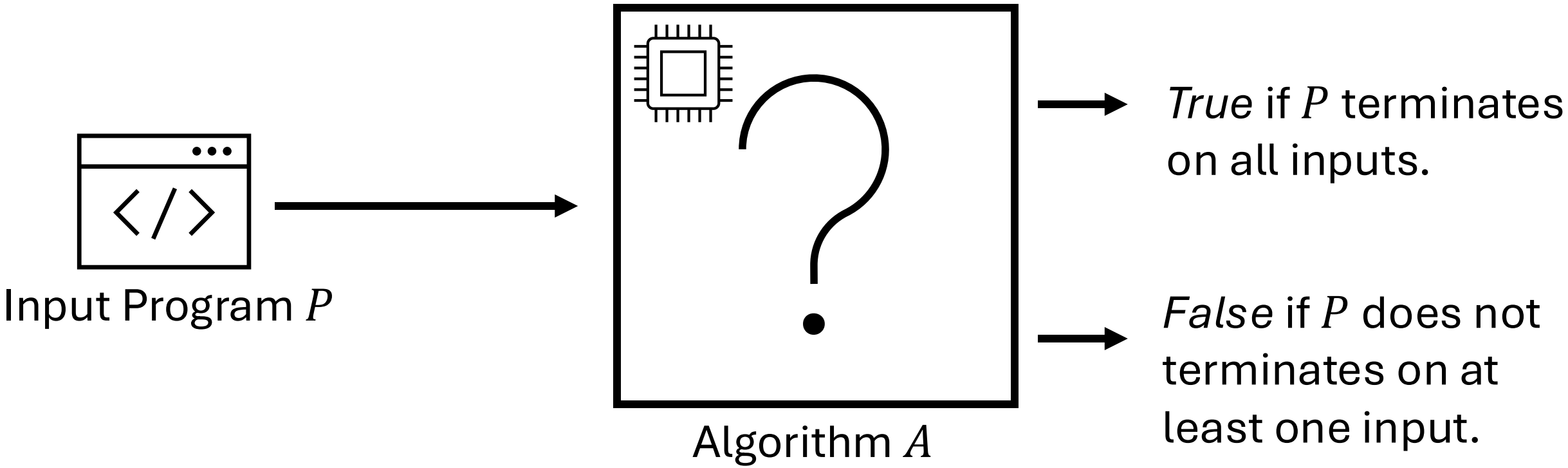
```
def add (a,b):
    c = a+b
    return c
```

```
def badd (a,b):
    c = a+b
    while (True):
        continue
    return c
```

```
def badd2 (a,b):
    c = a+b
    if (c == 2):
        return c
    while (True):
        continue
```

Does it halt?

Input: A Program P.
Output: True if P terminates on all inputs. Otherwise, false.

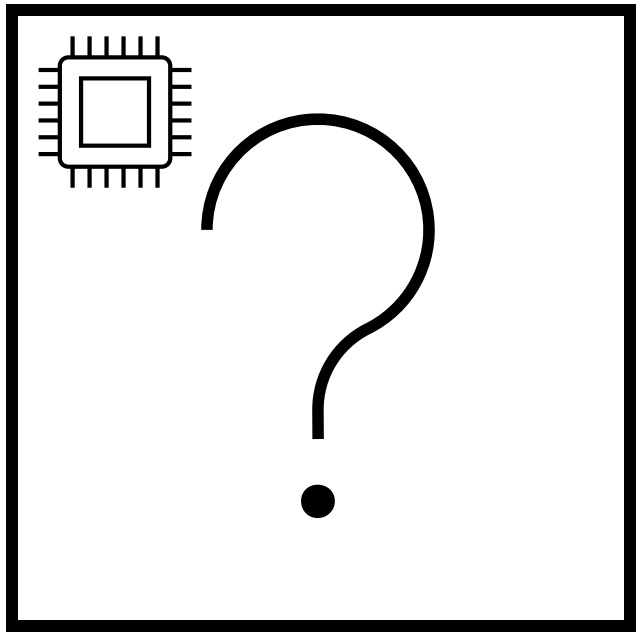


Does it halt?

Input: A Program P.
Output: True if P terminates on all inputs. Otherwise, false.

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

Input Program P



Algorithm A

True if P terminates on all inputs (a, b).

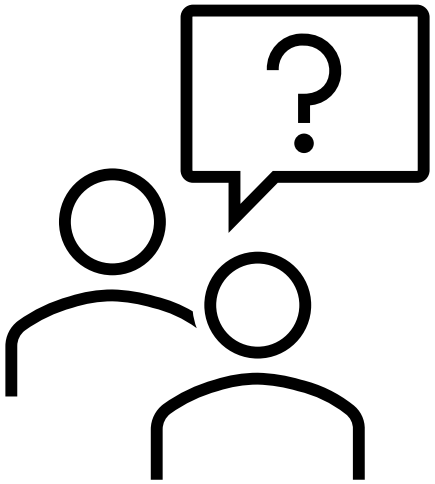
False if P does not terminates on at least one input (a, b).

Does it halt?

Q4: How would you try to solve this problem?

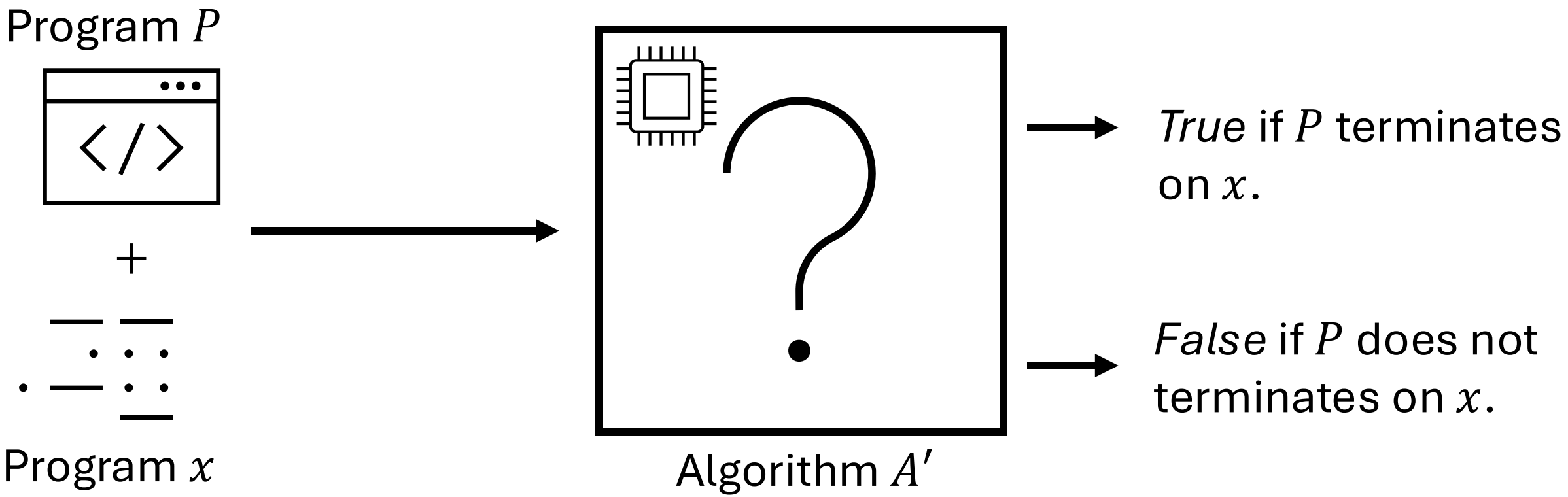
A4: You would not…

Q5: What could you do instead?



Does it halt 2.0?

Input: A Program P and Input x.
Output: True if P terminates on x. Otherwise, false.

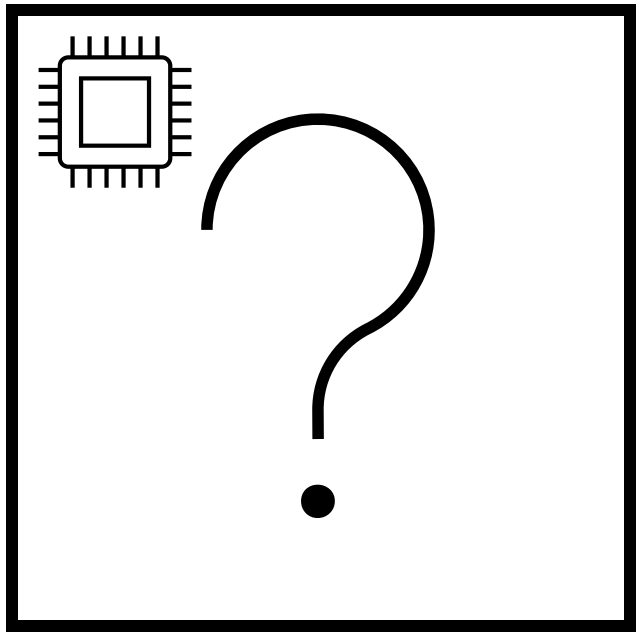


Does it halt 2.0?

Input: A Program P and Input x.
Output: True if P terminates on x. Otherwise, false.

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

Program P and input (a, b)



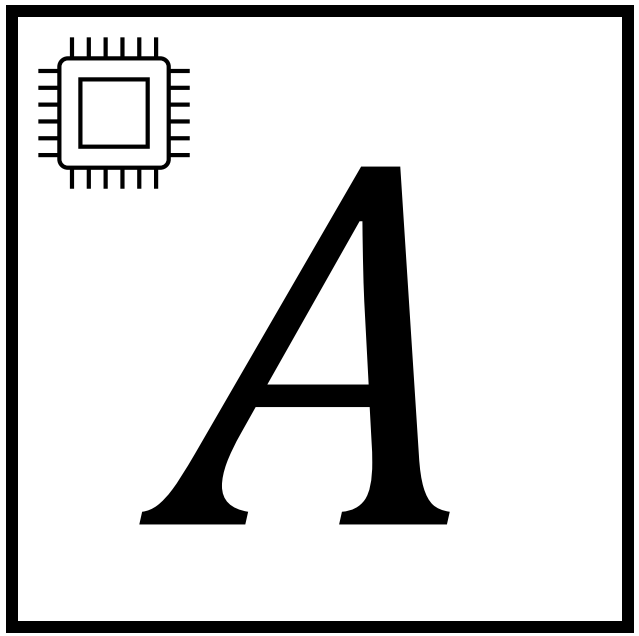
Algorithm A'

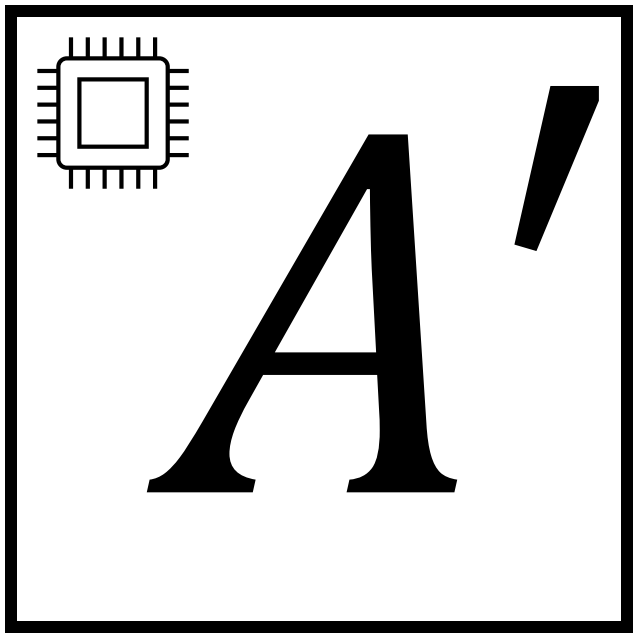
True if P terminates on (a, b).

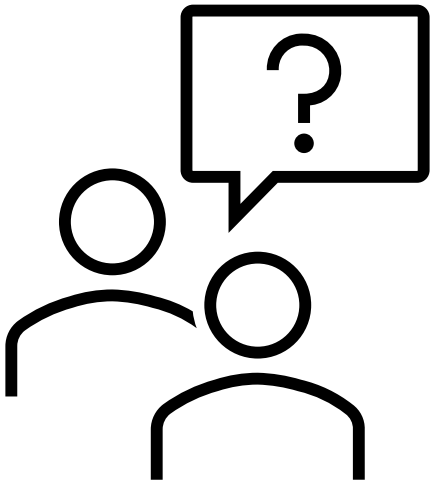
False if P does not terminates on (a, b).

Does it halt?

Q5: Which is harder to write? Why?

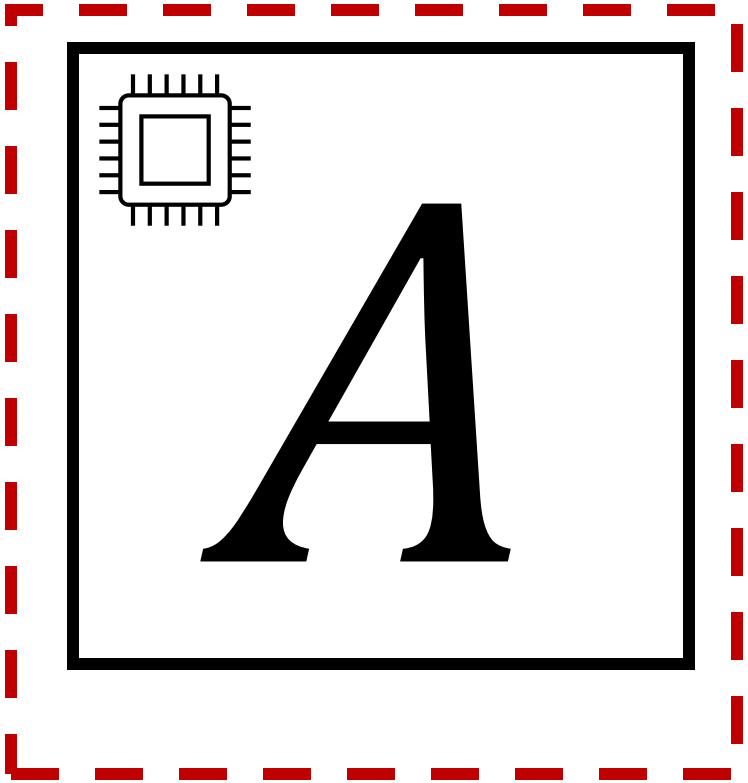


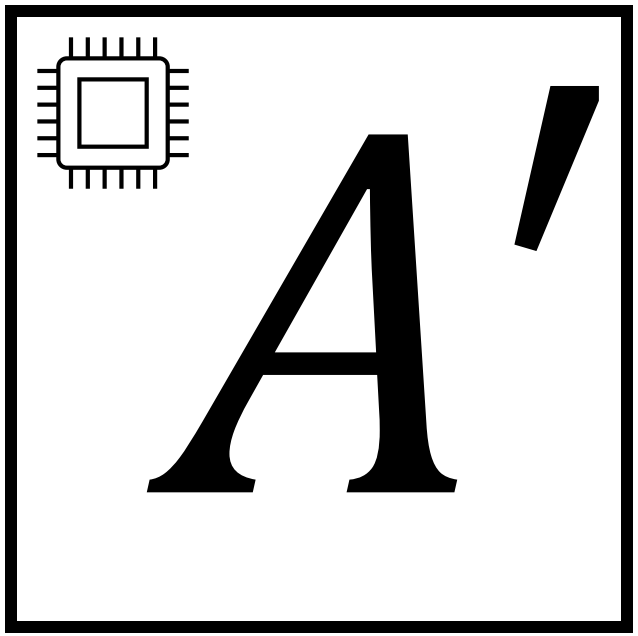


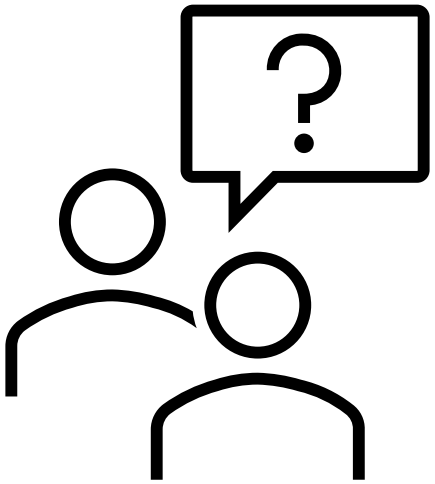


Does it halt?

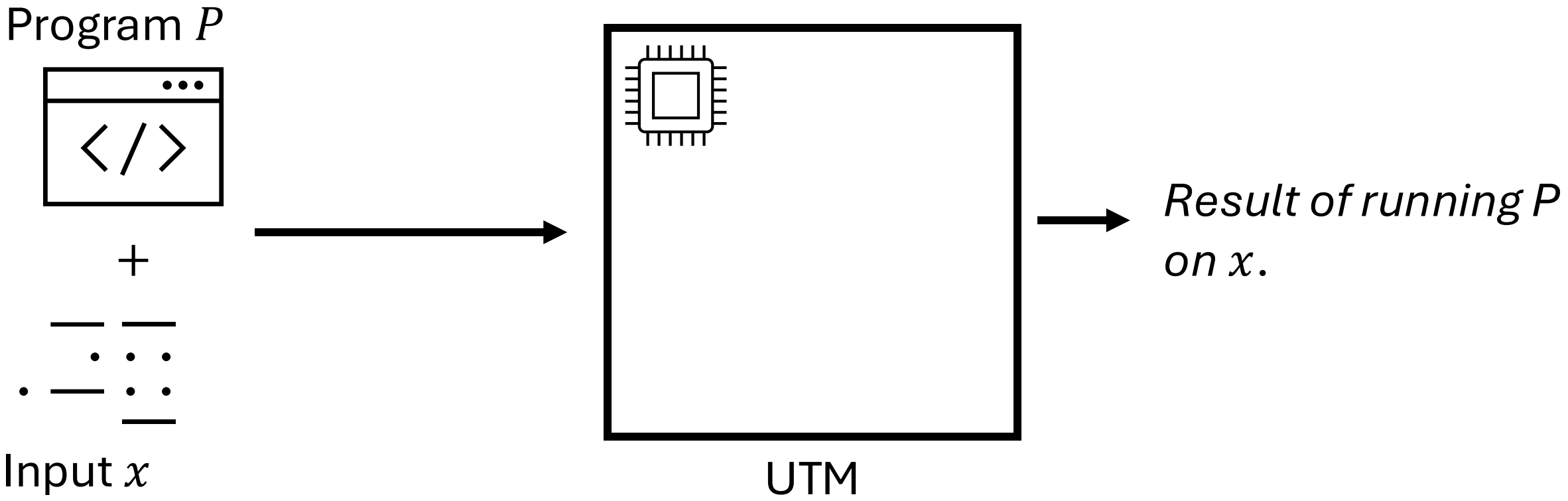
Q5: Which is harder to write? Why?



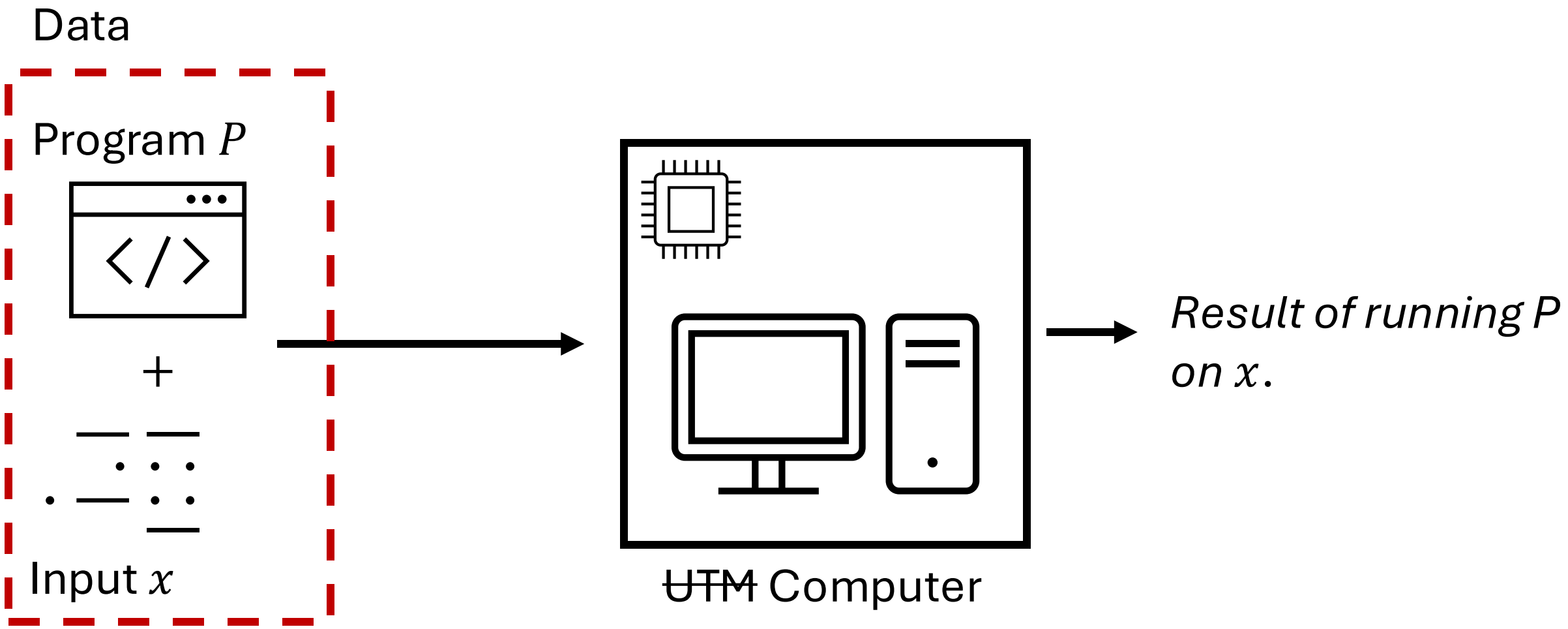




Universal Turing Machine



Universal Turing Machine



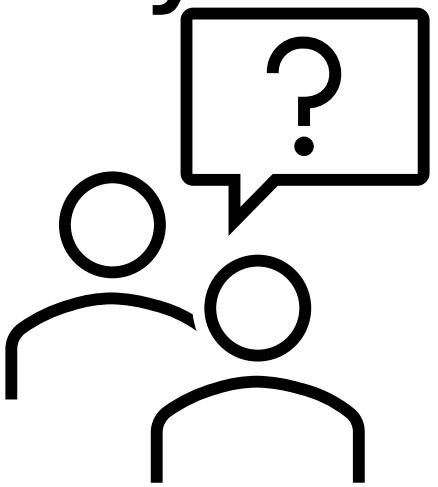
A Wild Theorem

Theorem: There exist no algorithm/program that will always halt and will always correctly decide Halting Problem 2.0.

A Wild Theorem

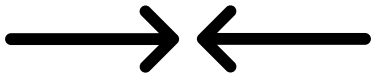
Q6: How would you prove this?

Theorem: There exist no algorithm/program that will always halt and will always correctly decide Halting Problem 2.0.



Proof by Contradiction

1. Assume negation of proposition.
2. Derive a contradiction from such an assumption.
3. Conclude that the original assumption is wrong and thus… proposition is true.



Theorem Proof

Suppose there exists an algo/program `MAGIC(P,I)` that decides if program `P` terminates on input `I`.

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

Theorem Proof

That means `MAGIC(P,I)` always terminates and returns the correct answer.

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

Theorem Proof

Define `Contradiction(P):`

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

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

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

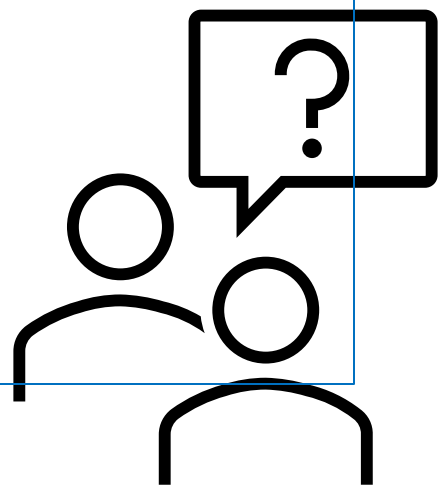
That's not right…

Q7: What if we run `Contradiction(Contradiction)` on the UTM?

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

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

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



Case I: P terminates on P

Q8: What happens if `Contradiction` terminates on `Contradiction`?

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

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

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

Case I: P terminates on P

Q8: What happens if `Contradiction` terminates on `Contradiction`?

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

#Run magic on (P,P)
if magic(P,P) True: # Use an UTM to make this call
    while True:
        continue # Do nothing
return # Just terminate if magic(P,P) returns False
```

Does not terminate!

Case II: P does not terminates on P

Q9: What happens if `Contradiction` does not terminate on `Contradiction`?

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

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

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

Case II: P does not terminates on P

Q9: What happens if `Contradiction` does not terminate on `Contradiction`?

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

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

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

Does terminate!

Proof Conclusion

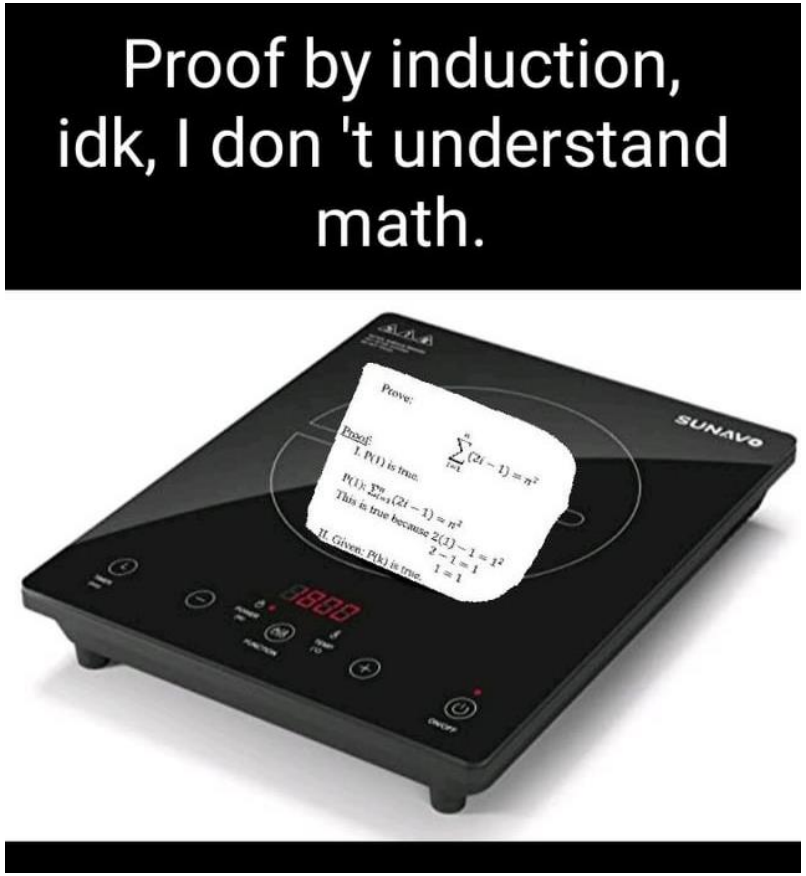
1. Assume `MAGIC` exists.
2. Use MAGIC to construct `Contradiction`.
3. "Law of excluded middle"
    1. If `Contradiction(Contradiction)` terminates then does not…
    2. If `Contradiction(Contradiction)` does not terminate then does…
4. Thus, something we said must be wrong.

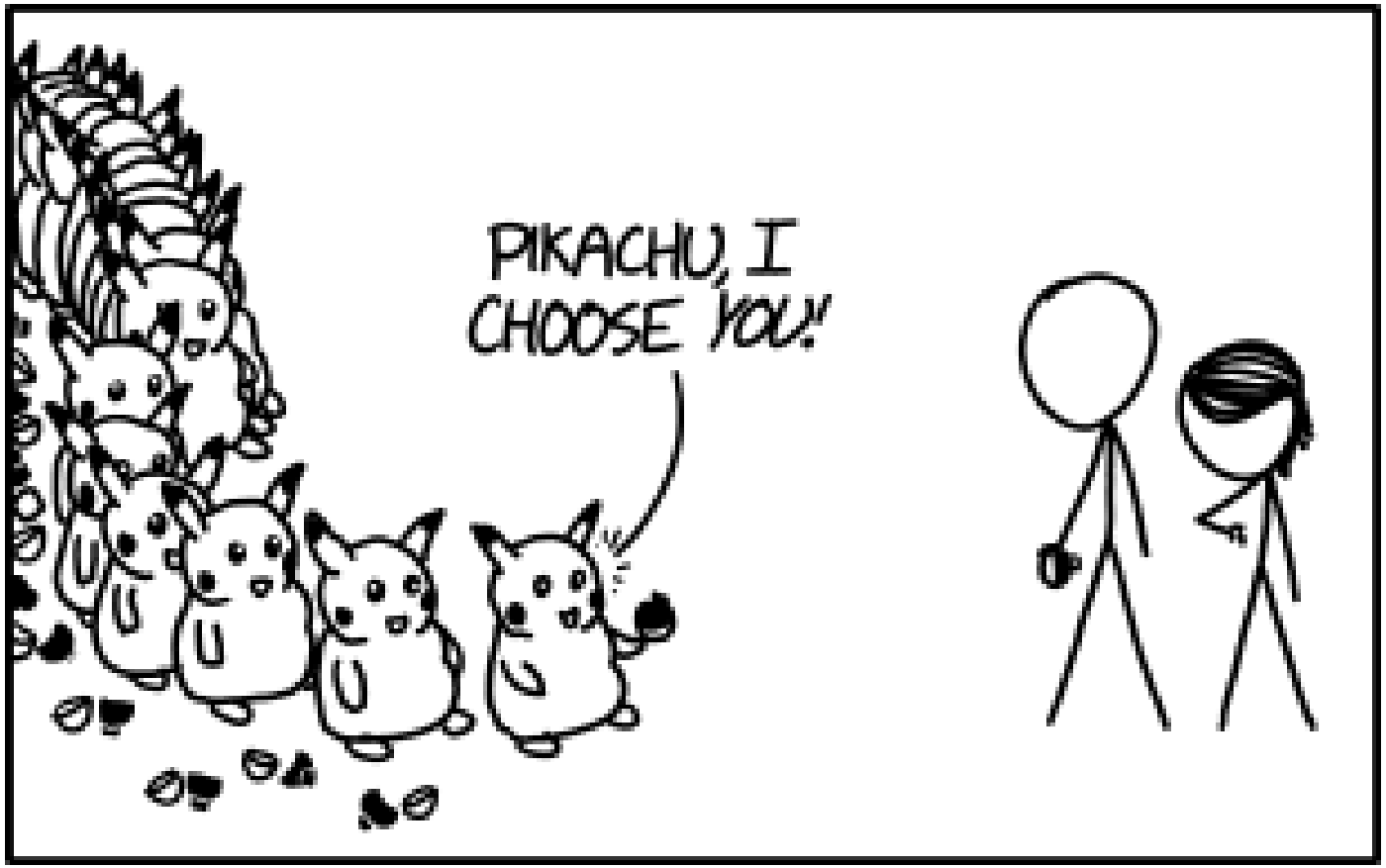
Q10: What could be wrong?

Questions?

Celebrate!

Proof by Induction





https://xkcd.com/1516/

Bongcloudforthewin. "**Proof by induction." Reddit, August 16, 2020.** **https://www.reddit.com/r/mathmemes/comments/iasdsb/proof_by_induction/**