CS 5 finale

Looking back! Evals, Ideas

CS 5: Final lecture now!

CS 5, on the verge of termination

I'll be back... that's my line!

Looking ahead? Options...

CS 5 Final Projects
due this Fri. eve (today!) weekend buffer, too
"Euro points" ~ likely have enough to skip hw12

CS 5 lab + office hours...
Today, 3-5 and 7-9
Saturday, 3-5
Sunday, 4-6

Worries? let's chat and/or drop me a note

CS 5 Final Exam
Not happening!
Final project ~ comments?!

one of many favorite grading - encountered comments

may you find equal success (and equal movie-watching time!)

# BWAHAHAHAHAHAHAHAHAHAHAHAHAHAH
# I AM THE WINNER AND IT ALL WORKS!!!
# GWHAHAHAHAHAHAHAHAH!!
# I started at noon, I'm done at 11:11
# minus a brief break to watch
# shawshank redemption and eat dinner,
# I've been working solid
# I AM TOTALLY THE WINNER!
# final project
The CS view of the world...

Input → Algorithm → Output

CS's challenge
Uncomputable functions

Sometimes an algorithm simply does not exist...

more precisely: every possible algorithm contains bugs!
Haltchecking is uncomputable.

\[ \text{hc}(f) \sim \text{return whether } f() \text{ halts or not} \]

\text{hc always has a bug!}
Haltchecking is uncomputable.

\[ \text{def } \textbf{hc}( f ) : \]

It is impossible to write a (bug-free) function \textbf{hc}( f ) that determines if a function \textbf{f} halts when run:

+ \textbf{hc}( f ) returns \textbf{True} if \textbf{f}() halts and
+ \textbf{hc}( f ) returns \textbf{False} if \textbf{f}() loops infinitely
Haltchecking is uncomputable.

\[ \text{hc}(f) \]

\[ \sim \text{returns whether } f() \text{ halts or not} \]

\[ \text{def } f1(): \]
\[ \text{ return 1 } \]

\[ \text{def } f2(): \]
\[ \text{ while 1+1==2: } \]
\[ \text{ print('Ha')} \]
\[ \text{ return 42 } \]

\[ \text{def } f3(): \]
\[ \text{ if f1() or f2(): } \]
\[ \text{ return 42 } \]

What boolean should \textbf{hc} return for each of these?

Extra! and what change - to only 1 character - would reverse the decision on \textbf{f3}?
Halt checking is uncomputable.

\[ \textbf{hc}( f ) \]

~ returns whether \( f() \) halts or not

\begin{align*}
\textit{def } f1() : \\
& \quad \text{return 1}
\end{align*}

\begin{align*}
\textit{def } f2() : \\
& \quad \text{while } 1+1==2 : \\
& \quad \quad \text{print('Ha')} \\
& \quad \text{return 42}
\end{align*}

\begin{align*}
\textit{def } f3() : \\
& \quad \text{if } f1() \text{ or } f2() : \\
& \quad \quad \text{return 42}
\end{align*}

\[ \textit{hc}(f1) \quad \textit{hc}(f2) \quad \textit{hc}(f3) \]

What boolean \textit{should} \textbf{hc} return for each of these?

Extra! What change - to only 1 character - would reverse the decision on \( f3 \)?
Suppose $hc(f)$ worked for all $f$. Create this `bff`:

```python
def bff():
    if hc(bff) == True:
        while 1+1==2: print 'Ha!'
    else:
        return  # halt!
```
Suppose $hc(f)$ worked for all $f$ 

Create this `bff`:

```python
def bff():

    if $hc(bff)$ == True:
        while 1+1==2:
            print 'Ha!'

    else:
        return # halt!
```

Is $hc(bff) == True$?

Is $hc(bff) == False$?

so, $hc$ always has a bug
Suppose \( \text{hc}(f) \) worked for all \( f \) 

Create this \texttt{bff}:

\begin{verbatim}
def bff():
    if hc(bff) == True:
        while 1+1==2: print 'Ha!'
    else:
        return # halt!
\end{verbatim}

Is \( \text{hc}(\text{bff}) \) == True ?

Is \( \text{hc}(\text{bff}) \) == False ?

so, \text{hc} always has a bug
And this is important because ...

∞ loops are undetectable

some are detectable, but some are not
– and there's no way to know!

bugs are inevitable

infinite loops are just one type of bug...
In general, they're all undetectable

programming is not automatable...

not perfect programming, at least
it's why CSers *celebrate* infinite loops!
it's why CSers *celebrate* infinite loops!

**Halting problem**

From Wikipedia, the free encyclopedia

In computability theory, the **halting problem** can be stated as follows: Given a description of a computer program, decide whether the program finishes running or continues to run forever. This is equivalent to the problem of deciding, given a program and an input, whether the program will eventually halt when run with that input, or will run forever.

**Alan Turing** proved in 1936 that a general algorithm to solve the halting problem for *all* possible program-input pairs cannot exist. A key part of the proof was a mathematical definition of a computer and program, what became known as a **Turing machine**; the
Corporate Address

Apple
1 Infinite Loop
Cupertino, CA 95014
408.996.1010

original iPhone icon for Maps

Apple Campus
One Infinite Loop
the iPhone's icon for Google Maps ... CS tourism!
the iPhone's icon for …
CS tourism @ ???...
CS tourism @ FB...
"Facebook is forever...!?"
Meaningful functions?

Input → Algorithm → Output

but, nearly all meaningful functions are computable...

the exception: functions that try to predict software behavior!
CS 5's examples...

Input ➔ Algorithm ➔ Output

Ciphertext ➔ Deciphering (via Caesar)

Two Dates ➔ The # of days between them

current gener. of “life” cells ➔ next gener. of “life” cells
CS 5's examples...

Input → Algorithm → Output

Ciphertext

Deciphering (via Caesar)

Two Dates

The number of days between them

Current generation of “life” cells

Next generation of “life” cells

Balance of utility & creativity / aesthetics?
"Science without religion is lame, religion without science is blind."
"Two things are infinite: the universe and human stupidity; and I'm not sure about the universe."
"Duct tape is like the force, it has a light side, a dark side, and it holds the world together"
"If you die in an elevator, be sure to push the Up button."
"All generalizations are false, including this one."
"Clearly you've never been to Singapore!"
"Luke, I am your father."
"To be, or not to be."
"You shall not pass!"
(... all with authors ...)
"Science without religion is lame, religion without science is blind."
"Two things are infinite: the universe and human stupidity; and I'm not sure about the universe."
"Duct tape is like the force, it has a light side, a dark side, and it holds the world together"
"If you die in an elevator, be sure to push the Up button."
"All generalizations are false, including this one."
"Clearly you've never been to Singapore!"
"Luke, I am your father."
"To be, or not to be."
"You shall not pass!"
(... all with authors ...)

"I have a dream! Duct tape is written on. Luke, I am your thoughts and what lies within us."
    ---- Audrey Rooney

"Your work is lame, religion is lame, religion is nearly the Up button."
    ---- Abraham Marx

"Two things are false, including this one."
    ---- Captain_Jack Truman

Markov-generated wisdom!
Compositional creativity

clay ~ computation...

Game of Life: GOL...!!!!

https://www.youtube.com/watch?v=xP5-ileKXE8
Compositional creativity

clay ~ computation...

balance of utility & creativity / aesthetics?
Allen Wu's cs5 memes!
Fortunately, nearly all *meaningful* functions are computable...

**but this doesn't mean we know how to compute them - yet!**
the computer vision problem
computer vision: what's the input and output?
Input $\rightarrow$ Algorithm $\rightarrow$ Output

2d array of ints

That's the Mona Lisa

(low difficulty)

That's a 16th c. woman, smiling maybe...

(high difficulty)
btw, why is it clear these data/image examples definitely don't match?

2d array of ints

an image ... of what?
a woman, smiling?
Happily, this problem is being solved...

Actual input: **pixels**

AIPOLY IS ABOUT TO RELEASE AN APP THAT HELPS THE BLIND "SEE" THROUGH THEIR SMARTPHONE.

but why is it so challenging?

Actual output: **contents**

a woman, smiling?

https://www.youtube.com/watch?time_continue=33&v=XMdct-5bERQ
What's red?

**Goal:** a coke-can collecting robot...
What's red?

not good enough...
Idea: Use hue

Increasing hue

hue = 25°

hue = -25°
What's red?

-25 < hue < 25

hue's too much!
Idea: Use hue

Increasing hue:
- hue = 25°
- hue = -25°
Idea: Use *hue* and *saturation*

- *Increasing hue*:
  - hue = 25°
  - hue = -25°

- *Increasing saturation*:
  - sat > 0.75
  - sat < 0.75
What's red?

-25 < hue < 25
saturation > 0.75

The door is still matched, too... why?
What's red?

in a sense, this is our own vision systems' fault...!

Aargh!

The door is still matched, too...  why?
Try it!

[1] Illusions? What are they in each of these four cases?
[2] What computations is your brain doing to cause them?
2 layers
2 layers
2 layers
2 layers
Are *these* four lines parallel?
Try it!

[1] Illusions? What are they in each of these four cases?

[2] What **computations** is your brain doing to cause them? **Why?**

Vision is more challenging than it might seem on first "glance"!
we don't always give our own vision system credit for *all the work* it's doing...
we don't always give our own vision system credit for *all the work* it's doing...
we don't always give our own vision system credit for *all the work* it's doing...

*If this feels "unreal"? What's the *most famous* pattern like this?*
the now-familiar striped dress...!

blue + black
vs.
white + gold
IS THE DRESS IN SHADOW?
If you think the dress is in shadow, your brain may remove the blue cast and perceive the dress as being white and gold.

THE DRESS IN THE PHOTO
If the photograph showed more of the room, or if skin tones were visible, there might have been more clues about the ambient light.

IS THE DRESS IN BRIGHT LIGHT?
If you think the dress is being washed out by bright light, your brain may perceive the dress as a darker blue and black.
whether or not it can be explained in one picture...  *xkcd will try!*
Let it go!

Door == coke can?  *We'll work around it!*
The coke-can collector: *seeking*...
The coke-can collector: *seeking*...
The coke-can collector: *success*!
How good is CV?

with human help, pretty good!

www.youtube.com/watch?v=Oie1ZcyeqlM
How good is CV on inanimate things?
Deepfakes Are Getting Better, But They're Still Easy to Spot

so many pixels!

Actual output: contents people, walking...
CS 5: the past...

Functions & variables
Recursion
Representations (binary, ascii)
Circuit design & Hmmm
Loop.
Dictionaries
Objects and Classes
Computability

Future CS?

looks like broccoli to me!

Uncomputable functions
Principles of CS
CS 60
Logic & Computability

Data structures

SW Engineering & AI, Neural Networks, Computer Vision, Robotics
Eng 85, 155, 158 & CS 125, 131, 132, 134, 136

Compliers, Programming Languages, Networking,
Operating Systems, Computer Architecture

Theory of Computation, Advanced Algorithms
Math 167, 168 & CS 141, 142

Systems

Interfaces

Beyond classes?

Classes?

Theory

Interfaces

Beyond classes?

Classes?
I hope you’re having a great semester so far and have a fun spring break planned!

I never thought I'd say this, but thank you so much for forcing us to learn all of the different data structures in your CS5 class last year. A lot of my day today was spent deciding whether list of lists or dictionaries were more appropriate structures for a dataset I'm building (and I'm not even a programmer, so tell your students that saying they want to go into consulting isn't an excuse anymore)!

Regardless, it reminded me of the three-eyed alien.

I hope all is well,
Nathan
However you're CSing, enjoy!

but what if I'm nostalgic for CS 5 itself?
However you're CSing, enjoy!

consider grutoring for CS5 ~ next fall or beyond...

but what if I'm nostalgic for CS 5 itself?
No matter what road you choose, it's likely to be in binary...
Thank you for joining CS5!

See you here in the fall, for our summer-cs5 reunion!

Final Projects: due *Friday* evening...

Buffer: *Weekend, til Sun. evening*
The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".
The basic structure is fixed for the first phase of work extending to some point in July. Everyone is invited to contribute to the discussion of the second phase. Sussman is coordinator of "Vision Project" meetings and should be consulted by anyone who wishes to participate.

Goals - General

The primary goal of the project is to construct a system of programs which will divide a vidisector picture into regions such as likely objects likely background areas chaos.

We shall call this part of its operation FIGURE-GROUND analysis.

It will be impossible to do this without considerable analysis of shape and surface properties, so FIGURE-GROUND analysis is really inseparable in practice from the second goal which is REGION DESCRIPTION.
Subgoal for July

Analysis of scenes consisting of non-overlapping objects from the following set:

balls
bricks with faces of the same or different colors or textures
cylinders.

Each face will be of uniform and distinct color and/or texture.
Background will be homogeneous.

Extensions for August

The first priority will be to handle objects of the same sort but with complex surfaces and backgrounds, e.g. cigarette pack with writing and bands of different color, or a cylindrical battery.

Then extend class of objects to objects like tools, cups, etc.