

review

classes 1-3

numbers
strings
lists
tuples
for loops
while loops
if/else statements
files

class 4

files
dictionaries
functions
classes

class 5

classes
modules
Excel module

thinking algorithmically

amcgrew [10:38 AM]

“Yeah, it was easy to find solutions online but most of them weren’t things that I could really explain how to do. It’s great practice for thinking though how to do things step by step.”

Computers are infinitely stupid!

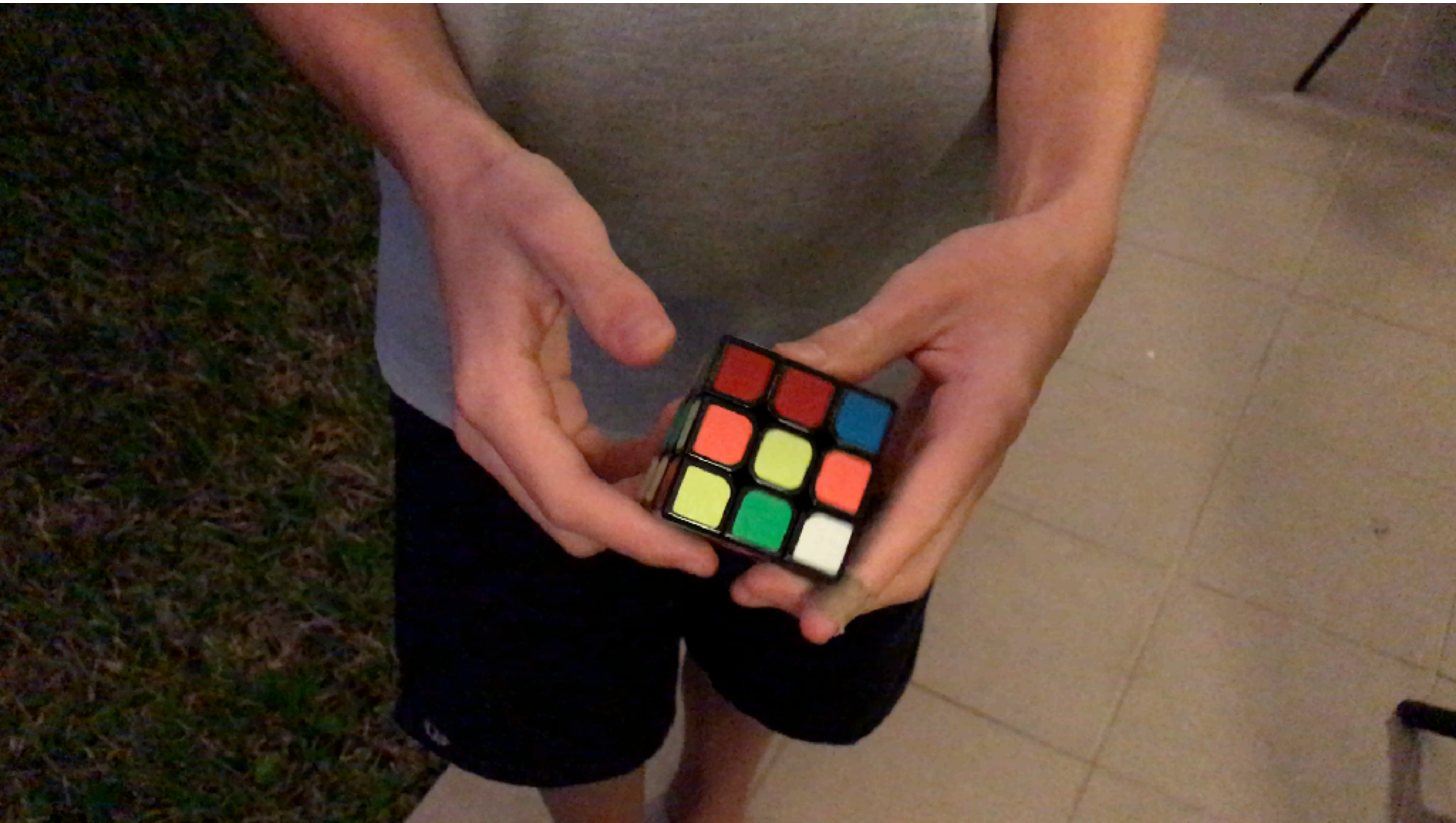
Computers only provide calculation power.

You provide the logic, reasoning, & intelligence.

You have to explicitly provide every detail of every step of a program. The computer cannot infer your logic.

Programming is most like puzzle solving.

thinking algorithmically



$$8! \times 3^7 \times (12!/2) \times 2^{11} = 43,252,003,274,489,856,000$$

Levinthal's paradox (1969)

Assume a protein consisting of only 100 amino acids

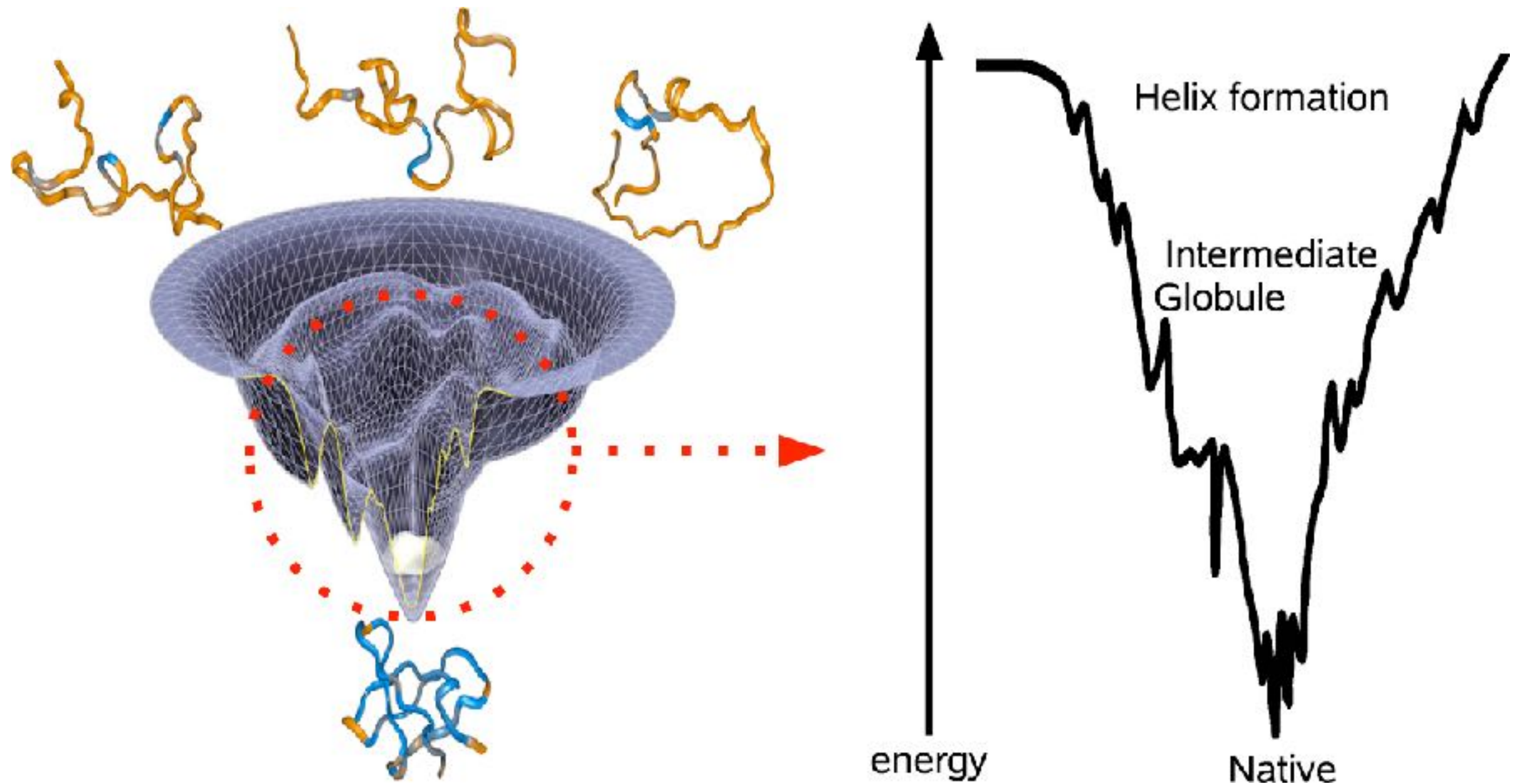
Assume that each amino acid can have 3 main chain rotations (in ϕ and ψ) — a very conservative assumption

Number of potential folds is then $3^{100} = 5 \times 10^{47}$

Typically, rotations take $\sim 10^{-13}$ s,
so it would take $\sim 5 \times 10^{47} \times 10^{-13}$ s = 1.6×10^{27} years
to search all conformational space

The age of the universe is 1.382×10^{10} years!

nature's folding algorithm



We are still not able to reliably fold proteins on computers!

try for general solutions

Why is this not a general solution?

```
stop_codon_index = []
for stop_codon in stop_codons:
    i = weirdAssDna.find(start_codon)
    while i < len(weirdAssDna) - 5:
        Codon = weirdAssDna[i:i+6]
        if Codon == stop_codon:
            if ((i + 6) - 31) % 6 == 0:
                stop_codon_index.append(i)
            i += 6
print("First stop codon", stop_codon, "found at index" , min(stop_codon_index))
```