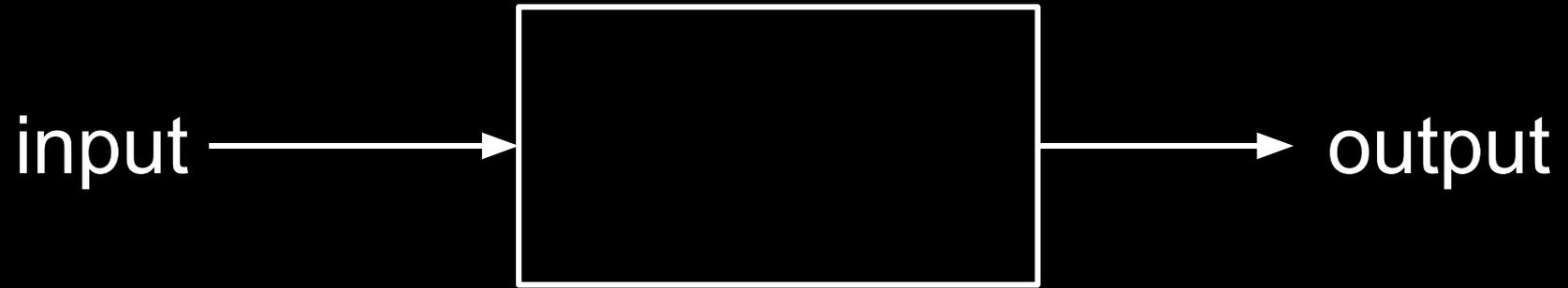
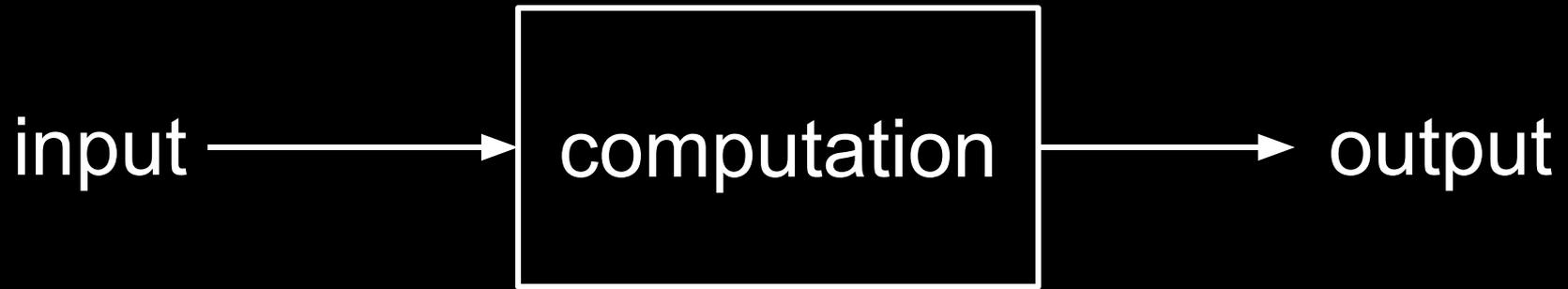


PROBLEMS

a model for solving problems



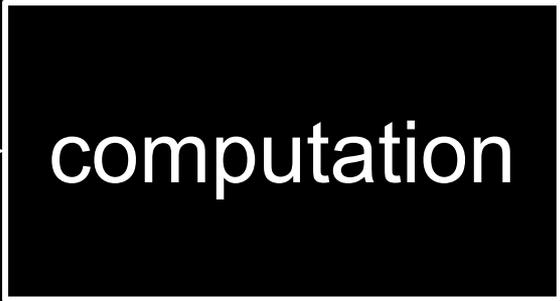
a model for solving problems



rules / instructions

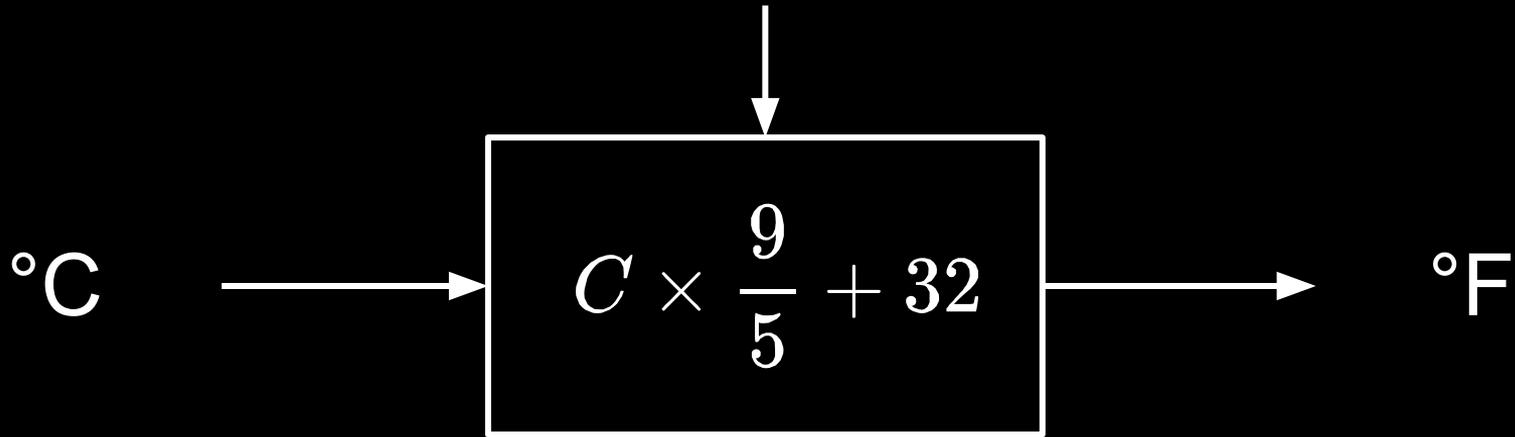


input

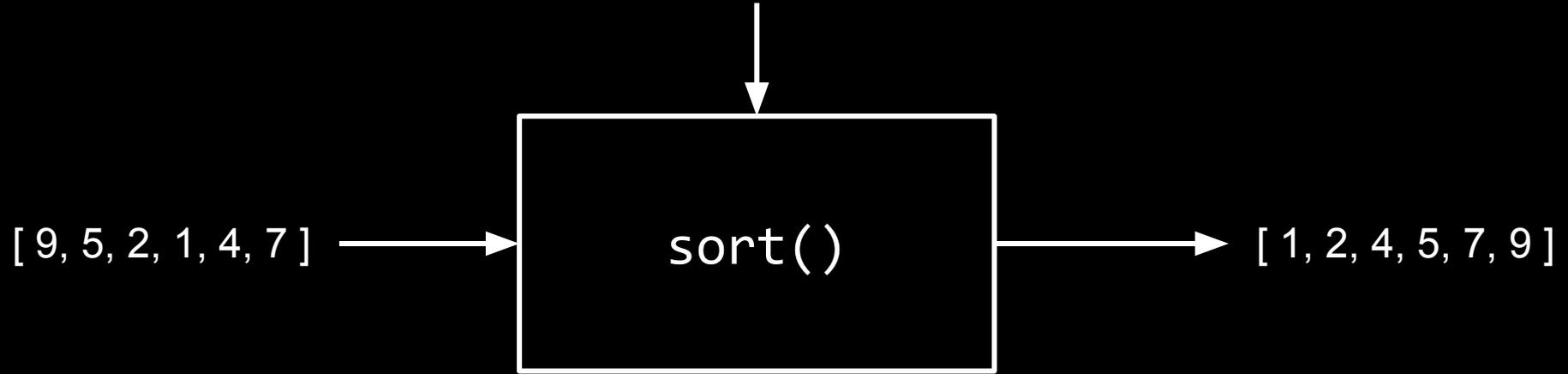


output

a simple rule



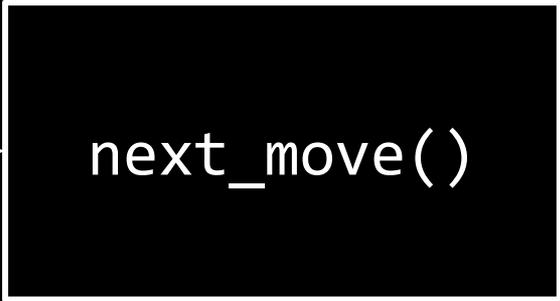
a set of instructions



selection sort:

find the smallest element and move it to front.

repeat for the rest of the elements.

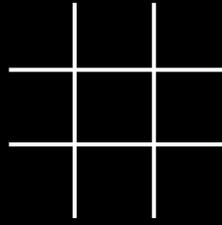


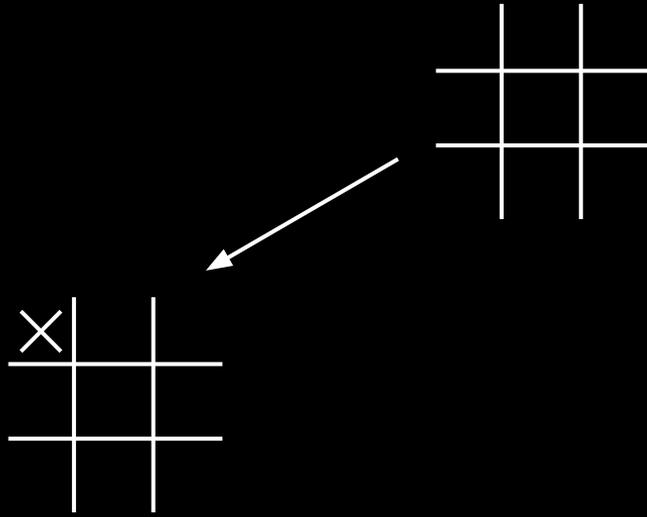
(2, 2)

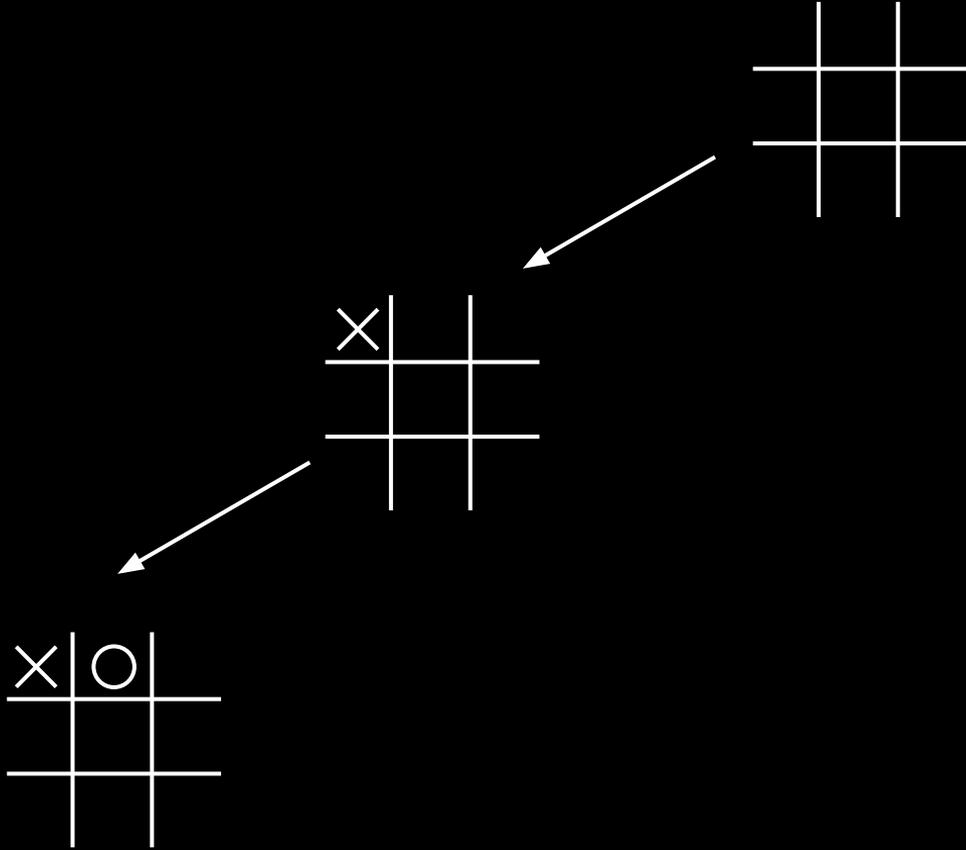
brute-force search:

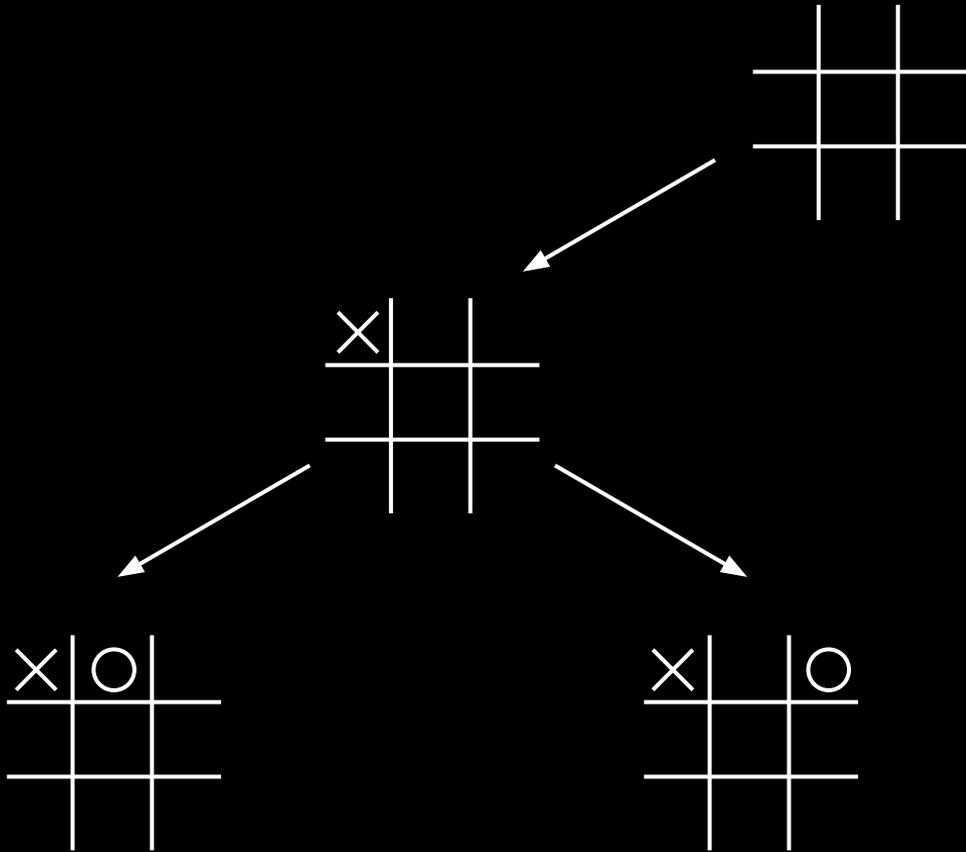
evaluate all possible move sequences.

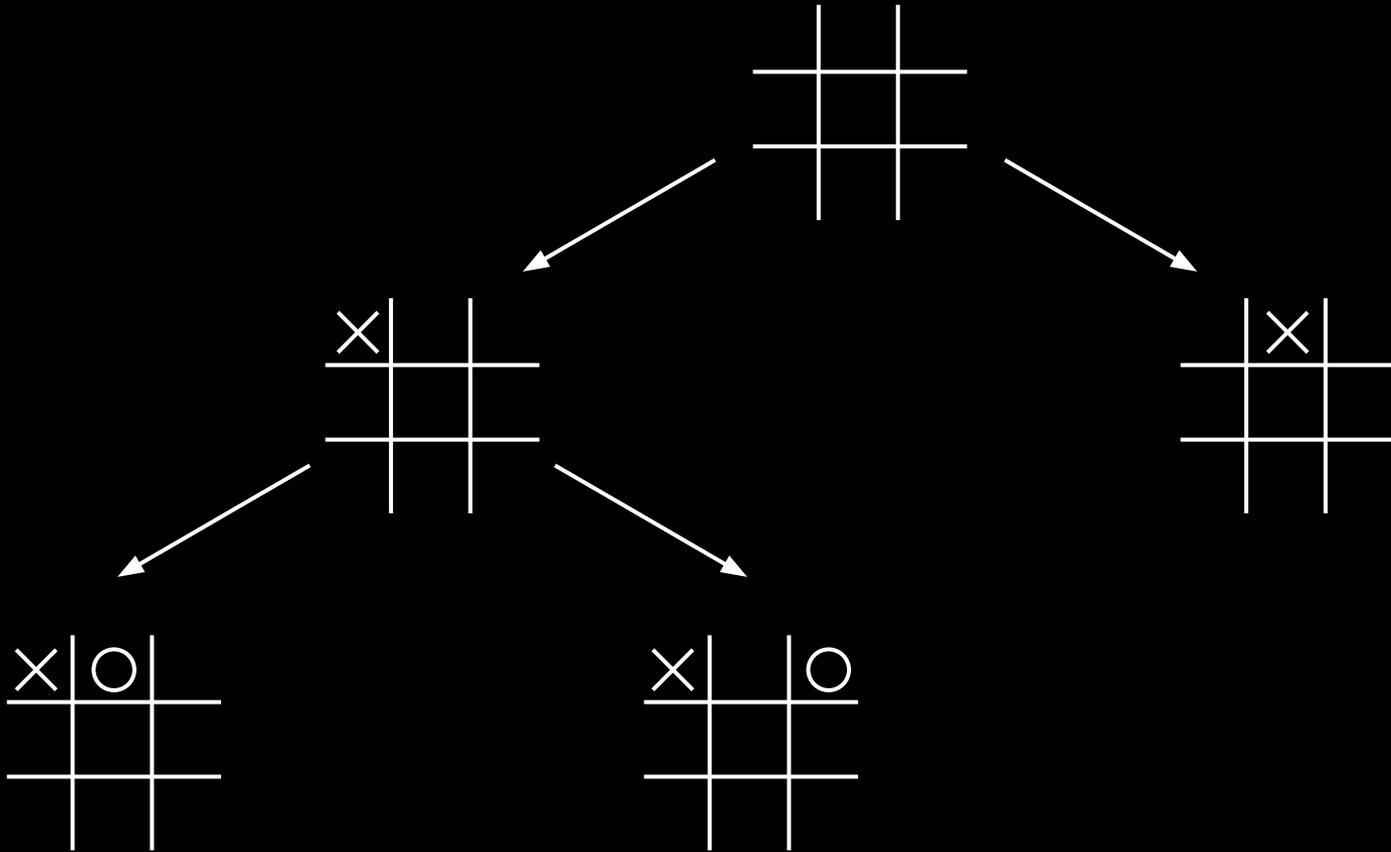
choose the move with the best value.

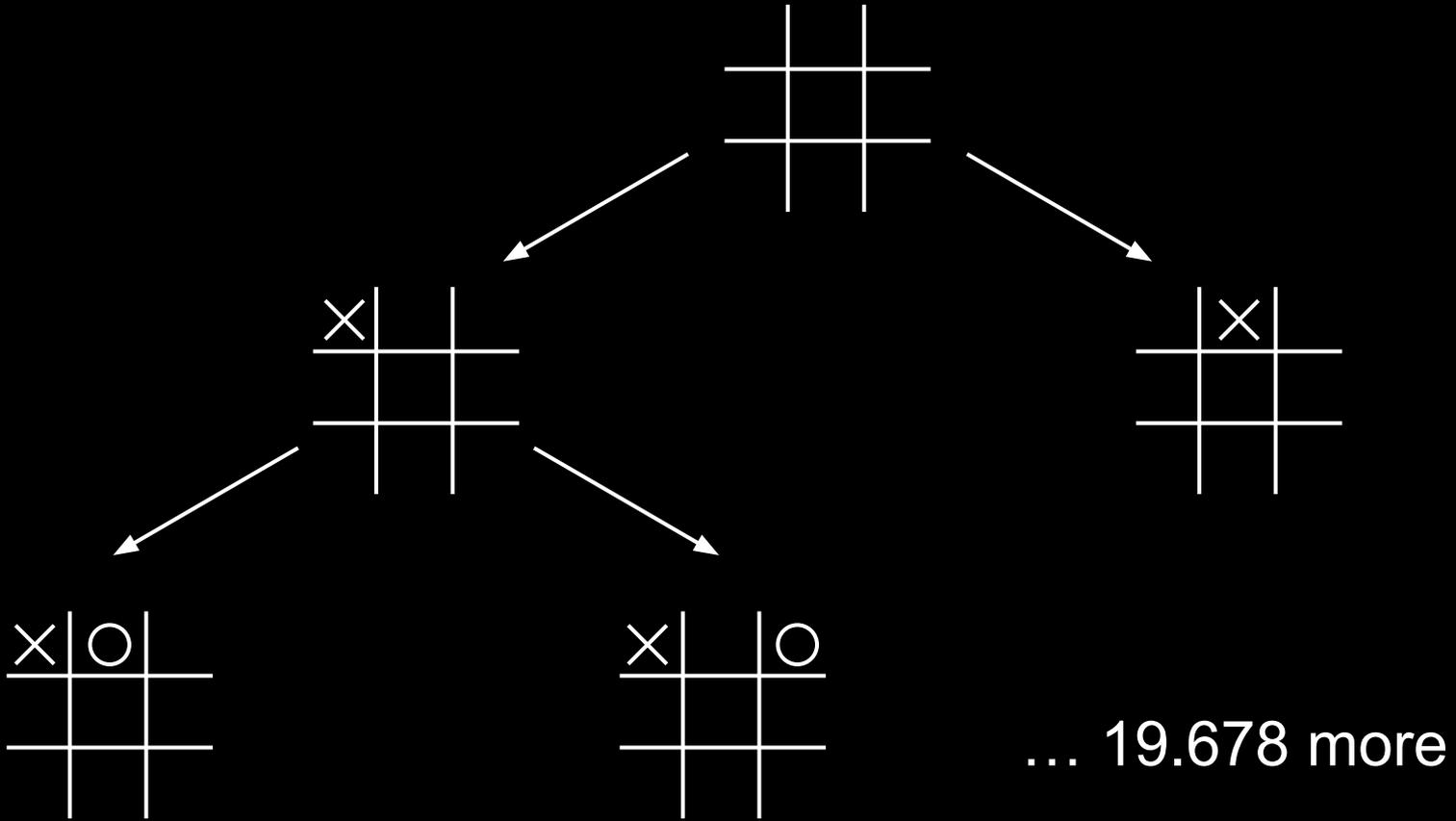




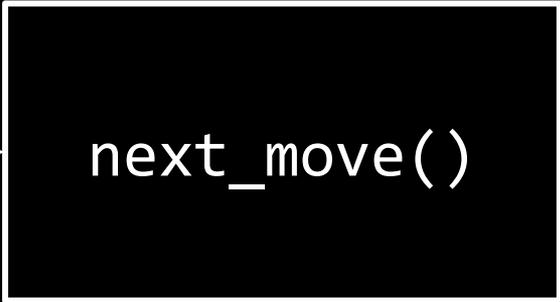






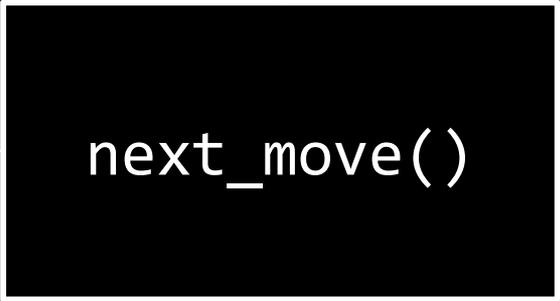


I. we can write rule-based programs to solve some problems

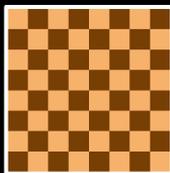


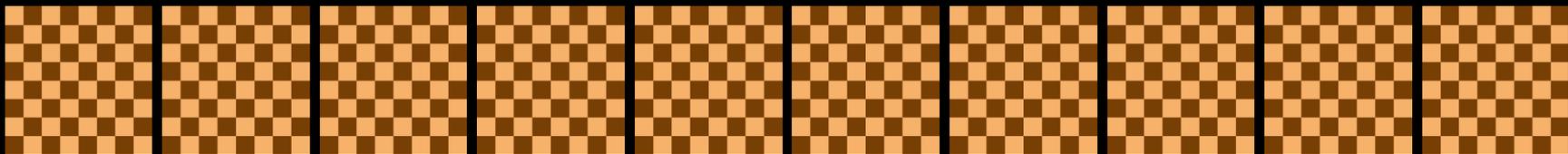
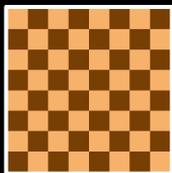
E2 → E4

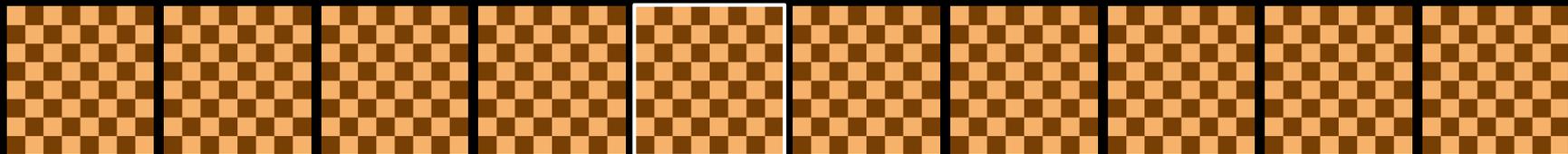
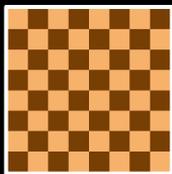
brute-force search ?

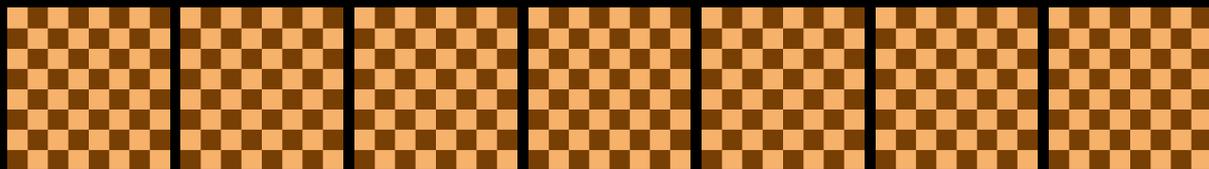
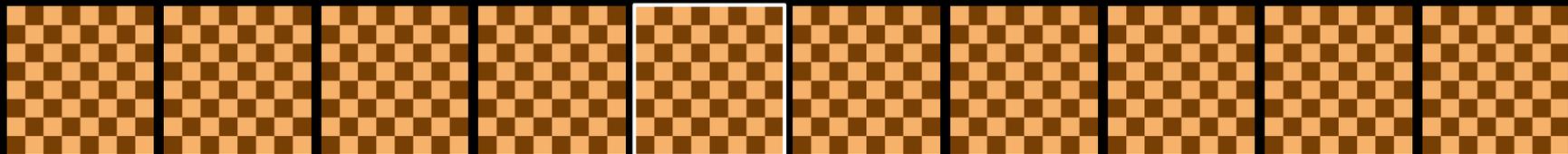
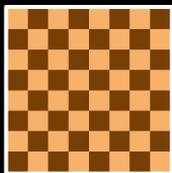


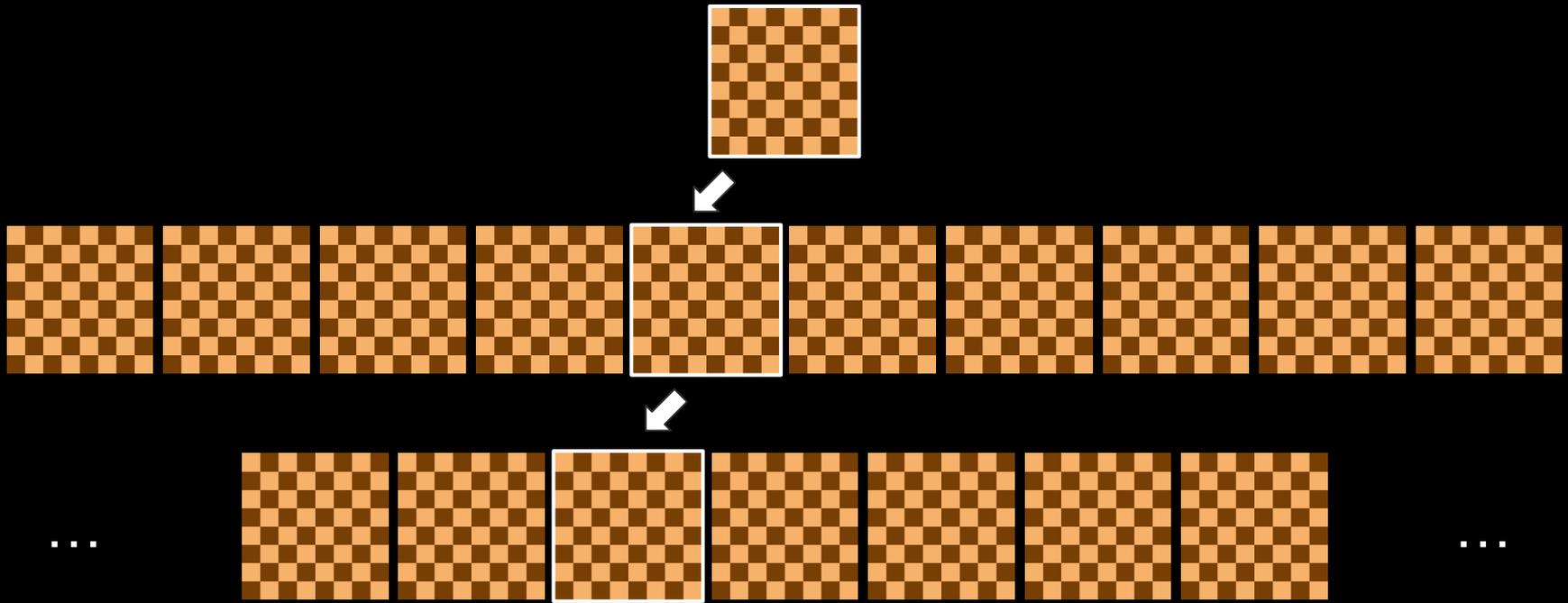
E2 → E4

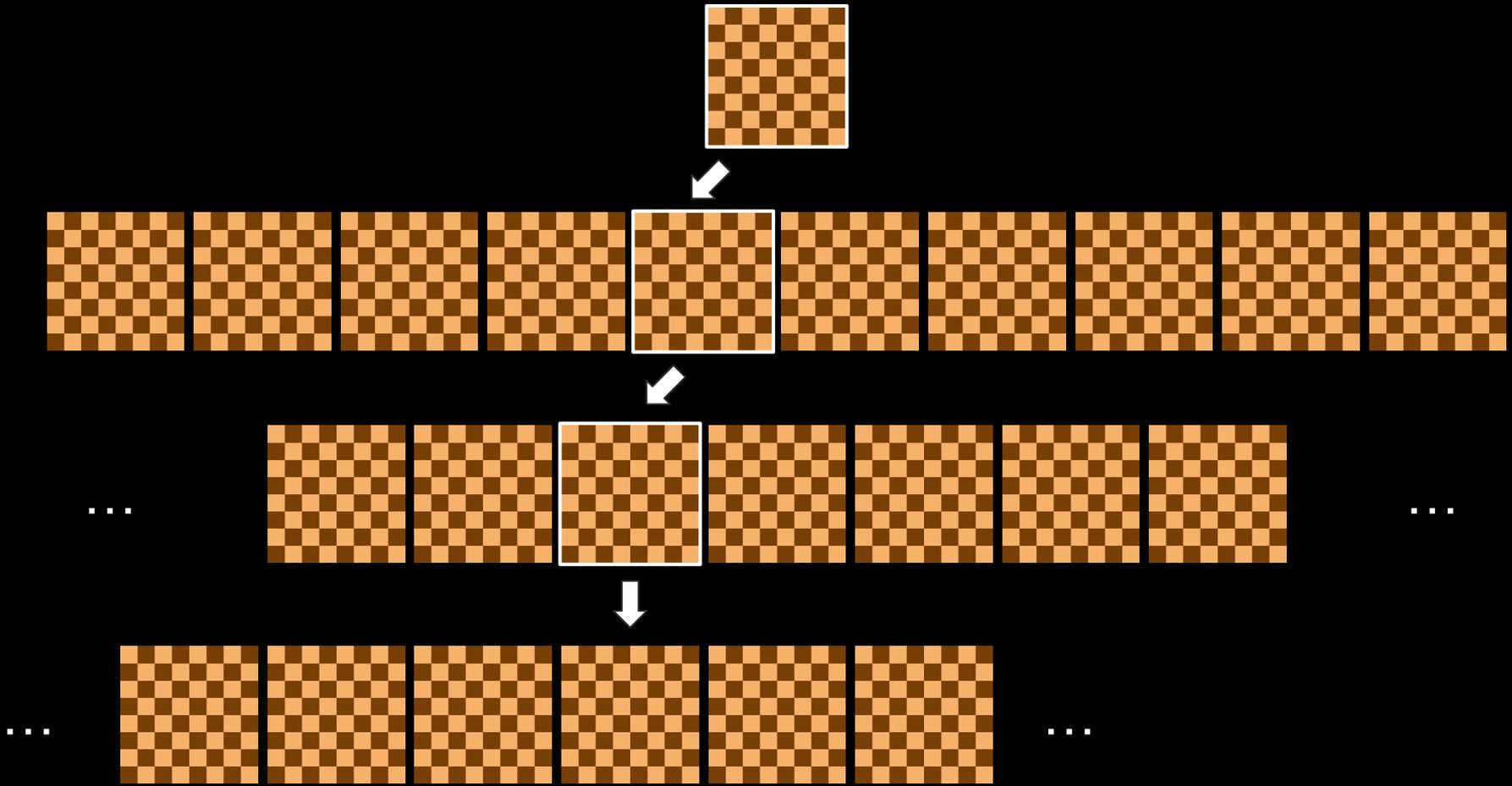


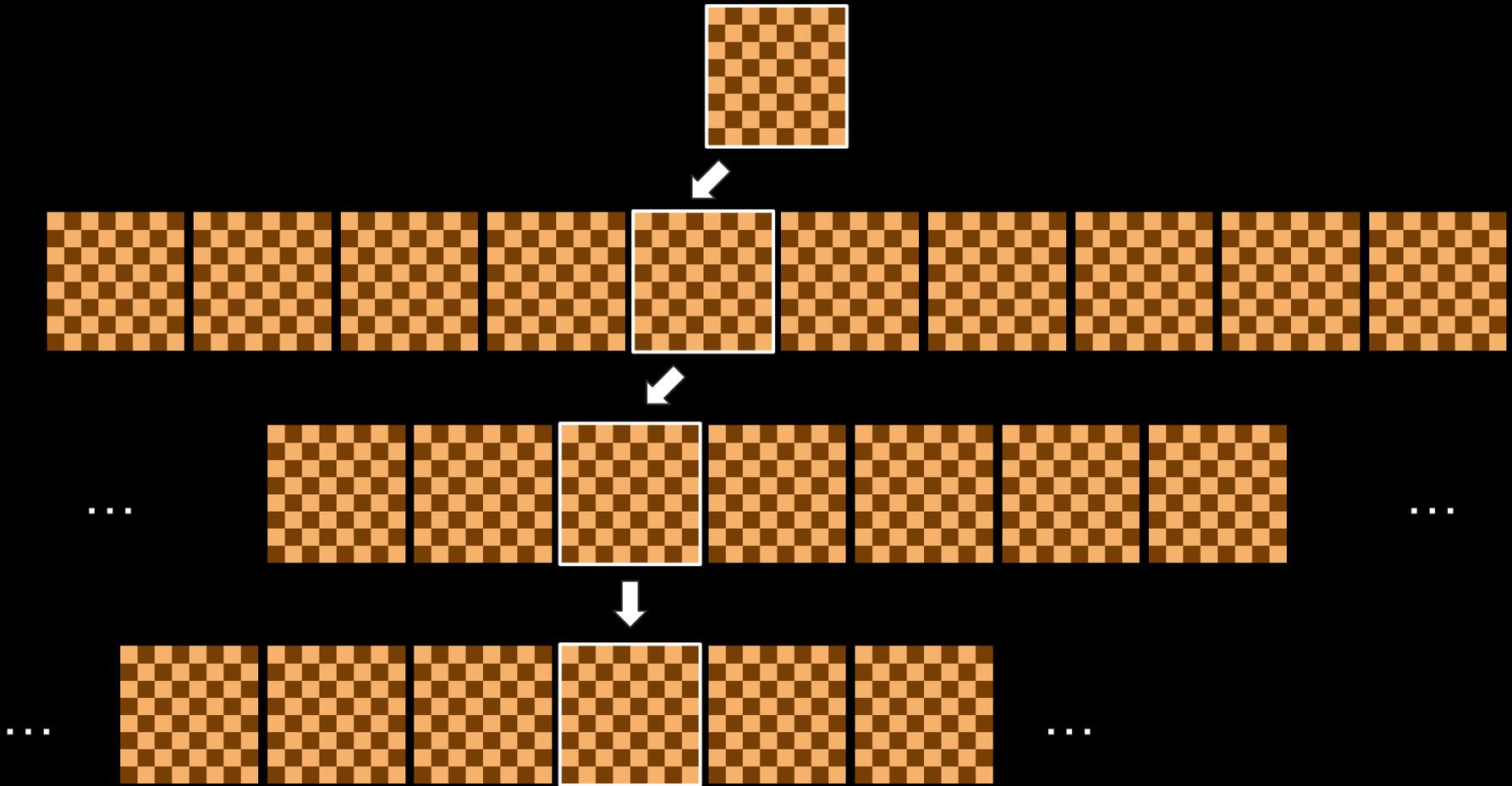












$\sim 10^{120}$ positions

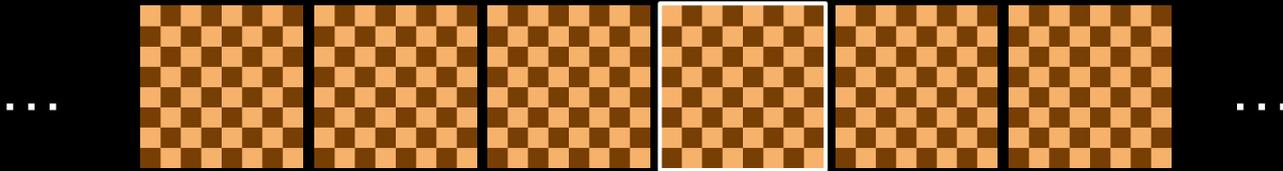
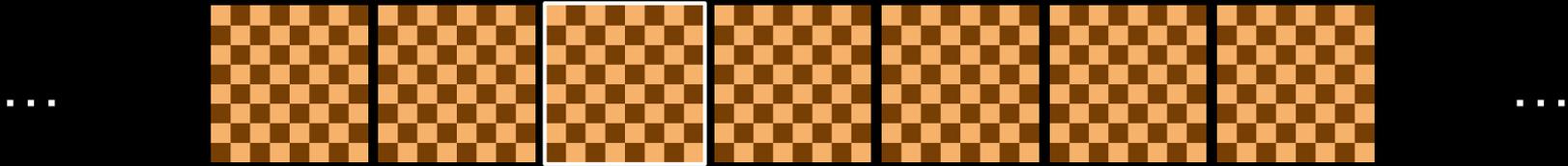
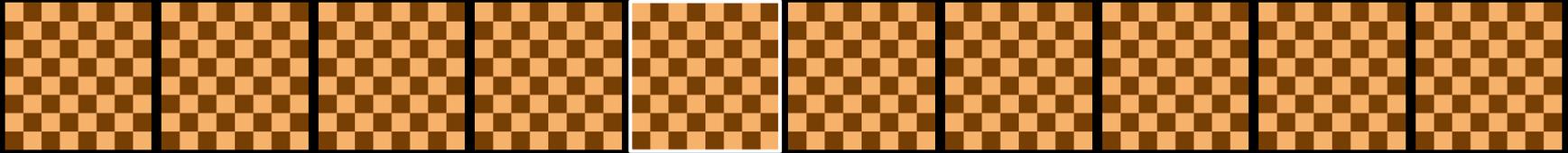
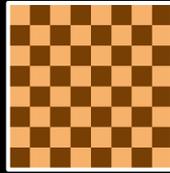




Image source: [IBM](#)



Image source: [IBM](#)

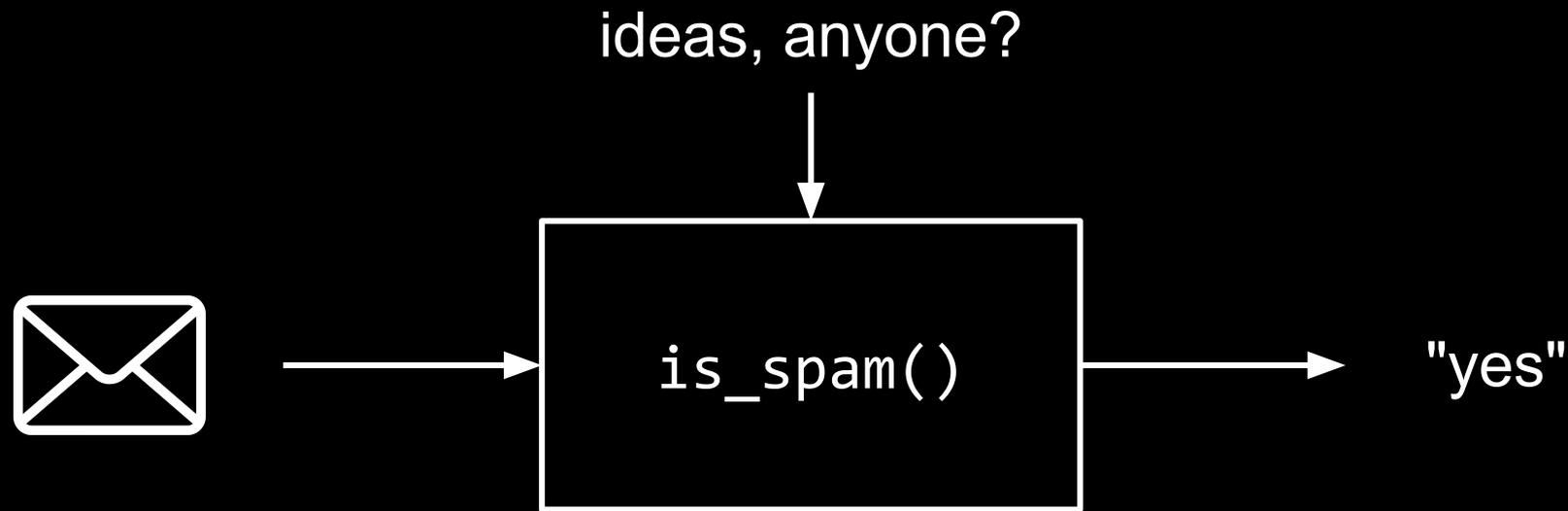
~ 200 billion positions per second

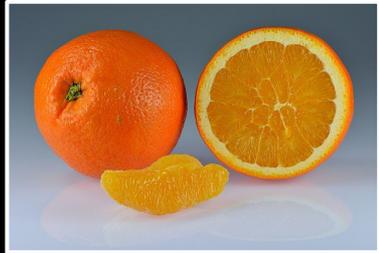
6-8 steps lookahead

- I. we can write rule-based programs to solve some problems
- II. some problems are too complex to be solved with rules alone



"yes"





`classify()`



"orange"

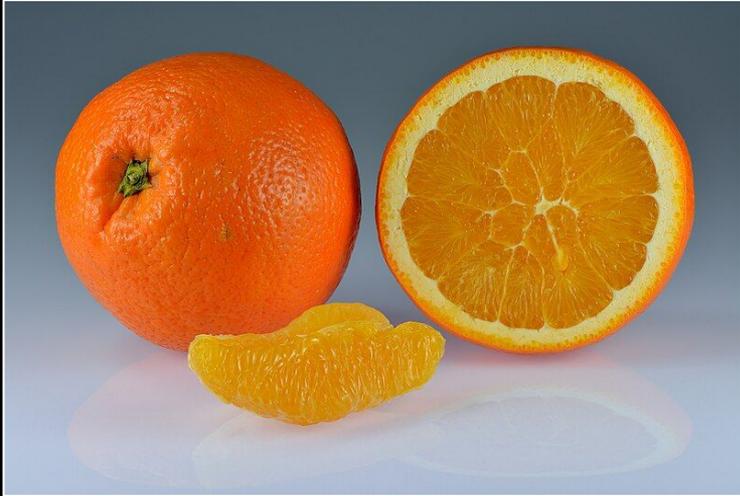


Image source: [Wikimedia](#)

rule-based attempt:

count all orange pixels.

```
if orange-ratio > 0.3:  
    return "orange"  
else  
    return "no orange"
```

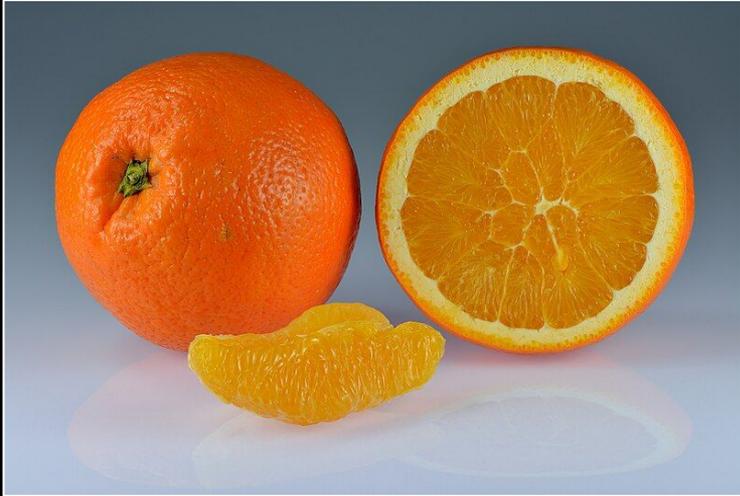


Image source: [Wikimedia](#)

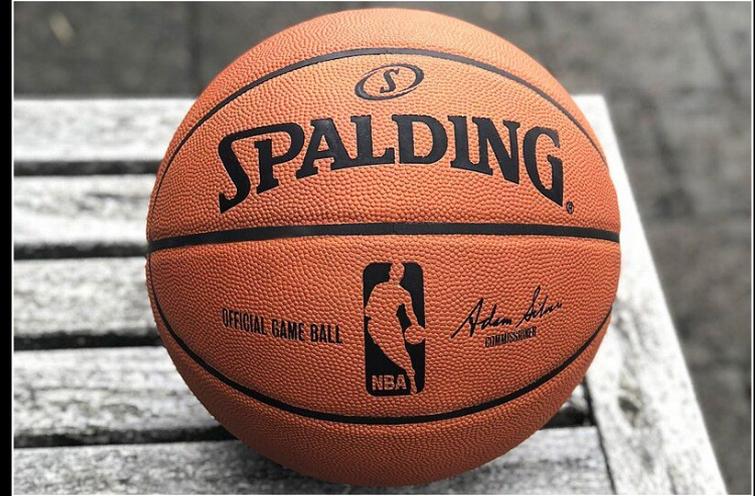


Image source: [Wikimedia](#)

Class 1 

Webcam 

HD Pro Webcam C920 (046d:082d) 



Zum Aufnehmen halten 

Bildbeispiele hinzufügen:

Class 2 

Bildbeispiele hinzufügen:

 Webcam  Hochladen

 Klasse hinzufügen

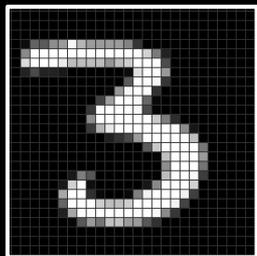
Training

Modell trainieren

Erweitert 

Vorschau  Modell exportieren

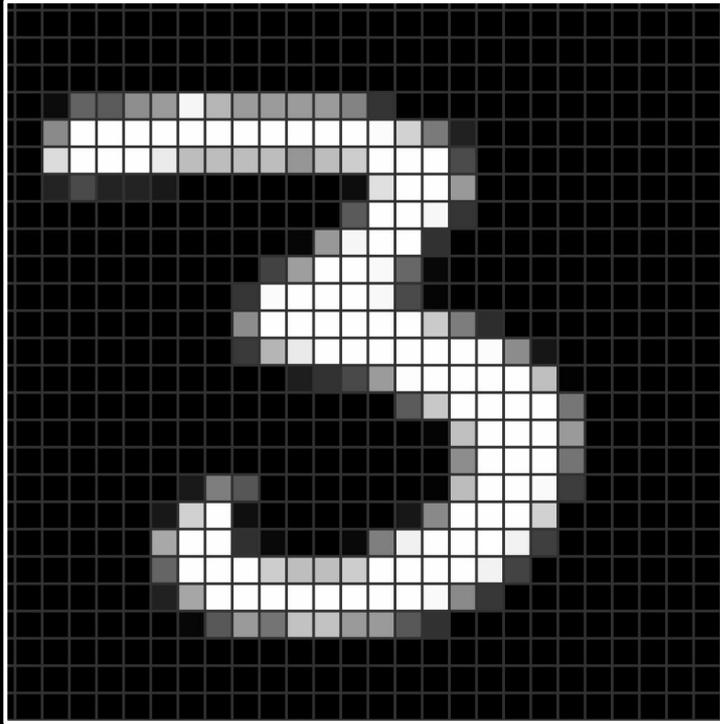
Du musst links ein Modell trainieren, um es hier als Vorschau ansehen zu können.



recognize()



3



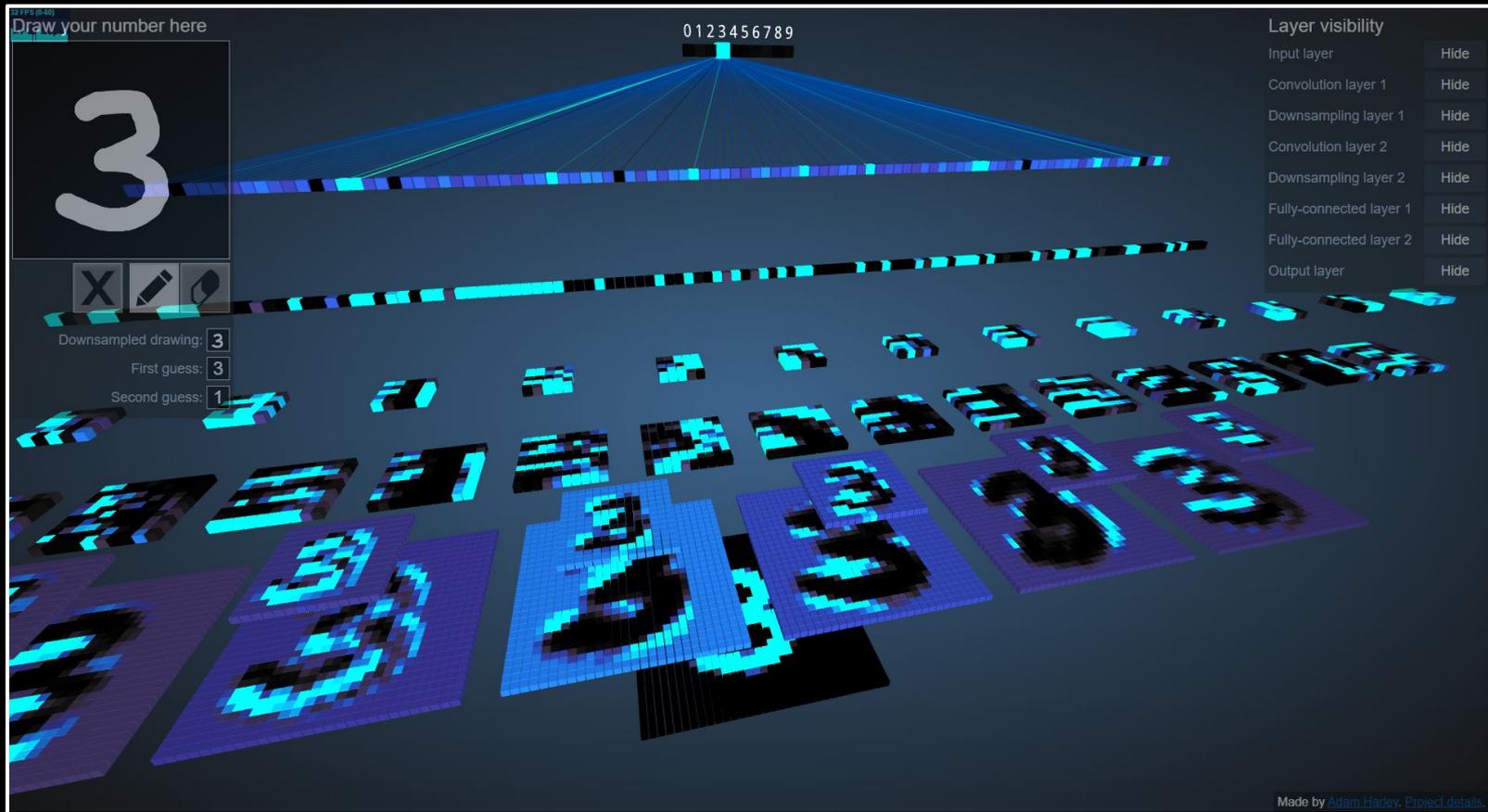
rule-based attempt:

define a list of pixel positions for digit 3.

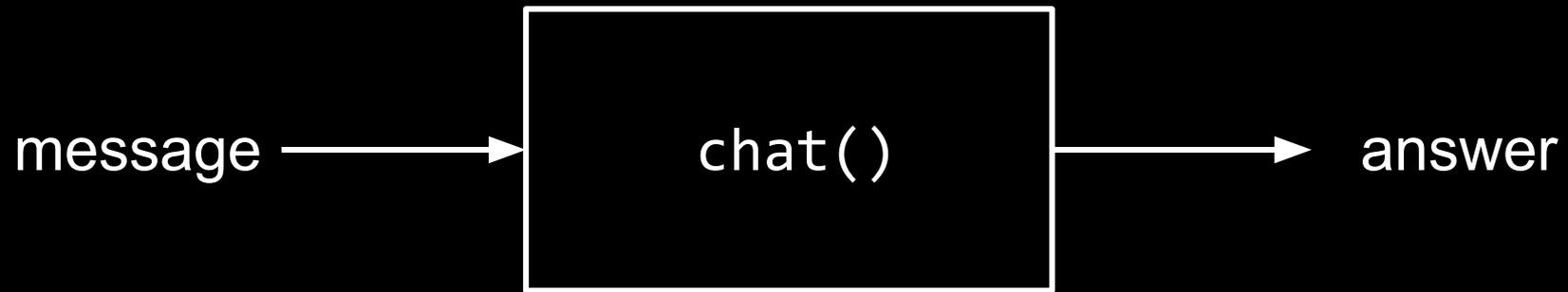
check each pixel and see if they match.

life is messy

3 4 2 1 9 5 6 2 1 8
8 9 1 2 5 0 0 6 6 4
6 7 0 1 6 3 6 3 7 0
3 7 7 9 4 6 6 1 8 2
2 9 3 4 3 9 8 7 2 5
1 5 9 8 3 6 5 7 2 3
9 3 1 9 1 5 8 0 8 4
5 6 2 6 8 5 8 8 9 9
3 7 7 0 9 4 8 5 4 3
7 9 6 4 7 0 6 9 2 3



https://adamharley.com/nn_vis/cnn/3d.html



Computational Linguistics

A. G. OETTINGER, Editor

ELIZA—A Computer Program For the Study of Natural Language Communication Between Man And Machine

JOSEPH WEIZENBAUM
Massachusetts Institute of Technology, Cambridge, Mass.*

ELIZA is a program operating within the MAC time-sharing system at MIT which makes certain kinds of natural language conversation between man and computer possible. Input sentences are analyzed on the basis of decomposition rules which are triggered by key words appearing in the input text. Responses are generated by reassembly rules associated with selected decomposition rules. The fundamental technical problems with which ELIZA is concerned are: (1) the identification of key words, (2) the discovery of minimal context, (3) the choice of appropriate transformations, (4) generation of responses in the absence of key words, and (5) the provision of an editing capability for ELIZA "scripts". A discussion of some psychological issues relevant to the ELIZA approach as well as of future developments concludes the paper.

The object of this paper is to cause just such a re-evaluation of the program about to be "explained". Few programs ever needed it more.

ELIZA Program

ELIZA is a program which makes natural language conversation with a computer possible. Its present implementation is on the MAC time-sharing system at MIT. It is written in MAD-SLIP [4] for the IBM 7094. Its name was chosen to emphasize that it may be incrementally improved by its users, since its language abilities may be continually improved by a "teacher". Like the Eliza of Pygmalion fame, it can be made to appear even more civilized, the relation of appearance to reality, however, remaining in the domain of the playwright.

For the present purpose it is sufficient to characterize the MAC system as one which permits an individual to operate a full scale computer from a remotely located typewriter. The individual operator has the illusion that he is the sole user of the computer complex, while in fact others may be "time-sharing" the system with him. What is important here is that the computer can read messages typed on the typewriter and respond by writing on the same instrument. The time between the computer's receipt of a message and the appearance of its response is a function of the program controlling the dialogue and of

Weizenbaum, Joseph. "ELIZA—a Computer Program for the Study of Natural Language Communication between Man and Machine." *Communications of the ACM*, vol. 9, no. 1, 1966, pp. 36–45, <https://doi.org/10.1145/365153.365168>.



- I. we can write rule-based programs to solve some problems
- II. some problems are too complex to be solved with rules alone
- III. for some problems, we cannot define a suitable set of rules to solve it